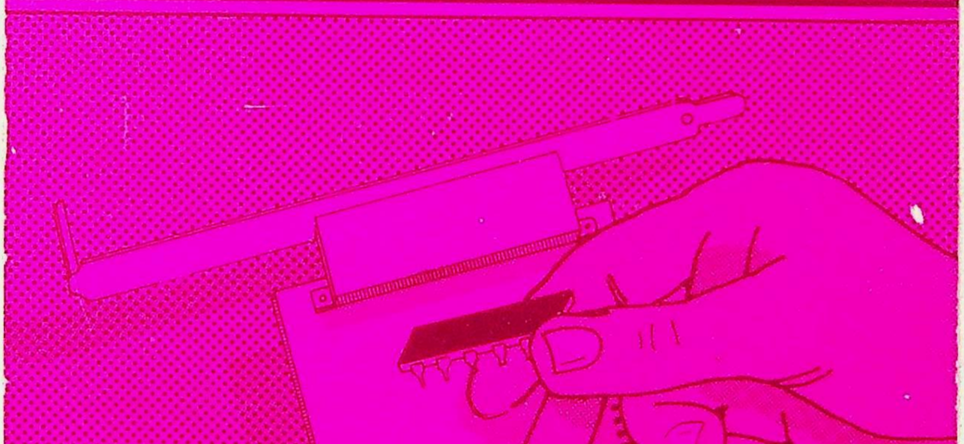
אריה נחום

יסודות המיקרו־מחשב

חומרה ורכיבים היקפיים



אריה נחום

יסודות

המיקרו-מחשב

חומרה ורכיבים היקפיים

1989 הוצאת אס.אי.אס.

תוכן עניינים

עמוד

עמוד

עמוד

עמוד

מבוא

ספר זה והניסויים שבו משלימים את ערכת התרגול בחומרה והיקפיים S-2100 של SES. יחד הן מהוות מערכת המאפשרת הוספת שלב בלימוד המיקרו-מחשב - לימוד ותרגול חומרה ורכיבים היקפיים של מיקרו-מחשבים. היא מאפשרת להכיר את הבעיות השונות הכרוכות בבניית החומרה, איתור תקלות והפתרונות השונים. אין אפשרות, כמובן, ללמד את כל אופני הבניה והפתרונות, אבל המטרה היא לתת בסיס מספיק רחב, שיאפשר ללומד להתמודד בעתיד עם דפי נתונים ורכיבים נוספים. הרכיבים הנלמדים בספר הם מהפופולריים בעולם ומהווים אבני בניה חשובות למדי.

שלב לימוד זה, בא לאחר שלב לימוד תוכנה של מיקרופרוססורים וכתיבת תוכניות בשפת מכונה. איפיון פתרון לבעיה במיקרו-מחשב כרוך בשילוב של חומרה ותוכנה. גם איתור תקלות במערכת מיקרו-מחשב, כרוך בכתיבת תוכנית מתאימה. זו הסיבה שאין אפשרות לתרגל חומרה ללא לימוד קודם של תוכנה למיקרופרוססורים. הנחת מוצא היא שהלומד מכיר את יסודות אחד המיקרופרוססורים 8085, 8051 או 8086 (ניתן ללמוד זאת באמצעות ספרים בהוצאת SES).

ערכת תרגול חומרה והיקפיים S-2100 מתחברת לערכת התרגול למיקרופרוססורים S-2085, S-2051 או ל-IBM PC עבור המיקרופרוססורים 8085, 8051 או 8086 בהתאמה. כתיבת התוכניות תתבצע על המערכות השונות ותרגול החומרה יתבצע על ערכת תרגול החומרה.

הערכה, מרכיביה ואופן השימוש בהם, מתוארים בשלבים בחלקים העיוניים של הניסויים בהתאם לשלבי השימוש בהם.

הספר כולל שמונה פרקים ובהם עשרים וששה ניסויים. בכל פרק חלק עיוני וחלק של תרגול מעשי. ניסויים בחומרה דורשים חיווט רב. ניתן לבצעם על לוח התקיעה שבערכה S-2100 או להיעזר בכרטיס DSM-2200, הכולל את מרבית הרכיבים הדרושים. כרטיס זה מאפשר ללומד לחסוך את שלב החיווט, הדורש זמן רב מזמן הניסוי.

ביצוע הניסויים בכל פרק יקח מספר שיעורי מעבדה. בכל ניסוי קיים חלק של כתיבת תוכנה. בחלקים העיוניים מוצעות תוכניות. התוכניות מופיעות כתרשימי זרימה או כתובות גם באסמבלי. על התלמיד להמיר תרשימים אלה לתוכניות, ולכתוב כתובות וצפנים. רצוי לבצע תרגילים בחומרה מעבר לניתן בספר. יש להקפיד על רישום מפורט ומסודר של כתיבת התוכניות, כדי להקטין למינימום את שלב איתור התקלות בניסוי עצמו.

ברוב הניסויים, התרגיל הוא בניית מערכת כלשהי, ובדרך כלל ללא תוצאות לעבוד. חשוב שהתלמיד יכין דו"ח מפורט הכולל תיאור הניסוי, מעגל חשמלי מלא, תוכניות שרשם בצורה מלאה כולל הערות ותיאור תוצאות הניסוי.

הכרת איפיון רכיב ואופן השימוש בו צריך להתבצע על-ידי קריאת דפי נתונים. חשיבות רבה יש להעמיק לנושא זה, עקב המיגוון העצום של רכיבים הקיימים במיקרו-מחשבים וצורות השימוש השונות בהם.

נשמח לקבל הערות והצעות לשיפור.

אריה נחום

פרק 1

תיאור כללי

1.1 מיקרו-מחשב - גישה אחרת

כל מערכת אלקטרונית בנויה מרכיבים רבים ושונים. בדרך כלל, מחולקים רכיבי המערכת לתת-מערכות שונות המתוארות בדיאגרמות המלבנים של המערכת. כל תת-מערכת מבצעת תפקיד מסוים וממלאת פונקציה מסוימת. גם מערכת מיקרו-מחשב בנויה מתת-מערכות שלכל אחת פונקציה מסוימת. תיתכן גם מערכת מיקרו-מחשב המבצעת פונקציה שמבצעת מערכת אלקטרונית רגילה ומבחינה חיצונית הן אפילו נראות אותו הדבר, כמתואר בציור.



ציור 1-1 סימול מערכת כלשהי

בזה מסתכם, למעשה, הדמיון בין מערכת אלקטרונית רגילה למערכת המשובצת מיקרו-מחשב ואפילו שתיהן ממלאות פונקציה זהה. לאחר תיאור המבנה הכללי והדינמיקה של מערכת מיקרו-מחשב, נשווה בין מערכת אלקטרונית רגילה ומערכת מיקרו-מחשב ונראה את המהפיכה בגישה ובדרך החשיבה שחולל המיקרו-מחשב לגבי מערכות אלקטרוניות המבוססות עליו.

1.2 מבנה כללי של מיקרו-מחשב

כל מיקרו-מחשב סטנדרטי בנוי מארבעה מרכיבים בסיסיים:

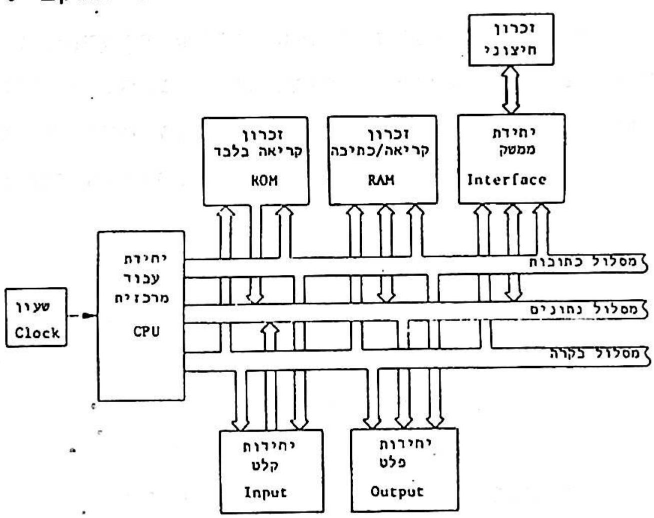
א. µP - מיקרופרוססור (מעבד זעיר) המהווה את ה-CPU (יחידת עיבוד מרכזית - Central Processing Unit) של המערכת.

ב. ROM - זכרון קריאה בלבד (Read Only Memory).

ג. RAM - זכרון קריאה/כתיבה (Random Access Memory).

ד. I/O - יחידות קלט/פלט (Input/Output).

ה-ROM, ה-RAM, וה-I/O מחוברים אל ה-CPU במקביל בצורה הבאה:



ציור 1-2 תיאור כללי של מערכת מיקרו-מחשב

המיקרופרוססור הוא יחידת עיבוד מרכזית הבנויה על פיסת סיליקון אחת. קיימות יחידות עיבוד מרכזיות רב פיסתיות, אבל אלה נמצאות בעיקר במחשבים גדולים. ליחידות עיבוד מרכזיות אלה יש בדרך כלל אורך מילה גדול יותר ומהירות גבוהה יותר. בספר זה לא נבדיל בין מיקרופרוססור ל-CPU, מכיוון שאנו עוסקים במיקרו-מחשבים שבהם המיקרופרוססור ממלא את תפקיד ה-CPU.

ה-CPU הוא לב המערכת, אליו קשורים כל שאר המרכיבים במקביל והוא זה שמפעיל אותם. ל-CPU כניסת שעון הקובעת את קצב המערכת. ברוב המיקרופרוססורים הקיימים היום, משולב השעון בתוך המיקרופרוססור ולאחרון יש יציאות לגביש הקובע את הקצב המתאים. השאיפה היא להגיע לקצב גדול ככל האפשר של פעולת המערכת. קצב זה מותנה במהירות המיתוג (זמני המעבר) של הרכיבים השונים. תדירות השעון היא פונקציה של מהירות התגובה של הרכיבים.

ה-CPU קשור אל שאר מרכיבי המערכת בשלושה מסלולים הנקראים BUS (כל מסלול נקרא BUS וזהו גם שם כולל לכל שלושת המסלולים).

<u>המסלולים הם:</u>

1. מסלול כתובות - ADDRESS BUS.
2. מסלול נתונים - DATA BUS.
3. מסלול בקרה - CONTROL BUS.

כל מסלול הוא למעשה אוסף של קוים המקשרים את ה-CPU אל מרכיבי המערכת. כל המרכיבים מחוברים אל ה-CPU במקביל כמתואר בציור. בספר זה נתייחס אל המיקרופרוססור בן 8 הסיביות בלבד. במיקרופרוססור זה המספרים המועברים בין ה-μP לרכיבי התמיכה שלו, הם מספרים בני 8 סיביות. אנו נעסוק גם ב-μP 8088 שהוא אמנם μP בן 16 סיביות, אבל גם אצלו המספרים המועברים הם בני 8 סיביות. כל מספר בן 16 סיביות מחולק לשניים ומועבר בשני שלבים. נרחיב את הדיבור על כך בהמשך, כמובן. מכאן שמסלול הנתונים מורכב מ-8 קוים שדרכם זורמים המספרים בני 8 הסיביות אל ה-μP וממנו.

מסלול הכתובות מורכב מ-16 קוים חד-כיווניים (מה-μP אל מרכיבי המערכת). בעזרת מסלול זה פונה ה-CPU אל המרכיבים

השרנים. 16 הקוים מאפשרים פניה בצורה ישירה ל-64K כתובות אפשריות, כאשר $K = 2^{10} = 1024$. ($2^{16} = 65,536$).
ב-8088 יש 20 קוי כתובת המאפשרים פניה ל-1M כתובות ($2^{20} = 1,048,576$).
מסלול הבקרה מורכב מ-10 עד 12 קוים בכיוונים שונים הנושאים אותות שונים בהתאם לתפקידיהם. בין היתר מאפשרים סינכרון המערכות השונות אל המערכת. מספר קוי הבקרה משתנה ממיקרופרוססור אחד למישנהו בהתאם לגישת היצרנים והדגשים שלהם.

כאמור, מסלול הנתונים הוא מסלול דו-כיווני. החיבור המקבילי של רכיבי התמיכה אל מסלול הנתונים מחייב שכל רכיב תמיכה יכיל במוצאו חוצצים שיאפשרו ניתוקו ממסלול הנתונים. אסור ששני רכיבי תמיכה יוציאו מידע אל מסלול הנתונים, כי הם יעמיסו זה את זה והמידע ישתבש. דבר זה מחייב שמערכת המיקרו-מחשב תכיל מעגלי פענוח שיחברו בכל פעם אל מסלול הנתונים, רק רכיב תמיכה אחד, על מנת שזה ימסור או יקבל מידע, בהתאם לפניית ה-CPU . מעגלי הפענוח צריכים באותה עת, לנתק את שאר הרכיבים ממסלול הנתונים - להעבירם למצב נתק (מצב 3-STATE).

בעזרת המסלולים, מעביר ה-CPU נתונים אל מרכיבי המערכת או מקבל נתונים מהם, כאשר הנתונים הם מספרים בינאריים בני 8 סיביות כל אחד. ה-CPU הוא זה שפונה אל מרכיבי המערכת ולכן מסלול הכתובות ורוב קוי הבקרה הם בשליטתו הבלעדית. קיימת גם אפשרות של העברת רוב יציאות ה-CPU למצב 3-STATE (מצב ניתוק) כדי לאפשר לגורם חיצוני השתלטות על המסלולים ולפנות ישירות למרכיבי המערכת ב-DMA (Direct Memory Access - גישה ישירה לזכרון). הגורם החיצוני יהיה עם מיקרופרוססור משלו.

ה-ROM הוא זכרון קריאה בלבד. המספרים הכתובים בו נקבעים בעזרת מערכת נפרדת למערכת המיקרו-מחשב ולכן אינם ניתנים

למחיקה או לכתיבה בעבודה רגילה של המיקרופרוססור. ה-ROM מכיל תוכניות קבועות ונתונים קבועים של המיקרו-מחשב שאינם נמחקים עם כיבוי המתח למערכת.

ה-RAM הוא זכרון קריאה/כתיבה. ל-CPU יש אפשרות לקרוא ממנו נתונים ולכתוב בתוכו נתונים. ה-RAM משמש כזכרון לתוכניות משתנות הנכתבות על-ידי המשתמש או שמסופקות למערכת המיקרו-מחשב מיחידה חיצונית וכן כזכרון לערכים המשתנים בזמן פעולת המערכת. עם כיבוי המתח למערכת, נמחקים הנתונים ב-RAM. עם חיבור המתח למחשב, מופיעים ב-RAM מספרים אקראיים.

יחידות ה-I/O הם רכיבים המשמשים את ה-CPU להתחברות אל העולם החיצון. העולם החיצון יכול להיות ציוד היקפי כמו זכרונות מגנטיים (הדורשים כוננים מיוחדים), מדפסות, צגים, לוחות מקשים וכד'.

למערכות בקרה, איסוף נתונים, מיתוג ופיקוד המופעלות על-ידי המיקרו-מחשב, משמשות יחידות ה-I/O כרכיבי ממשק (Interface). קיימות יחידות I/O הנקראות PIO (Programmable Input Output). רוב רכיבי ה-PIO למיניהם ניתנים לתיכנות בהתאם לתפקידיהם השונים. חלק גדול מספר זה מוקדש לחיבור המיקרו-מחשב לעולם החיצון ולהפעלת רכיבי PIO שונים.

1.3 <u>הדינמיקה של המיקרו-מחשב</u>

כל מערכת מיקרו-מחשב היא מערכת מסונכרנת. ה-CPU עובד בהתאם לקצב שעון (Clock) המחובר אליו או הנמצא בתוכו. בעזרת חלק מקוי הבקרה מוציא ה-CPU אותות, המסונכרנים לשעון, למרכיבים השונים של מערכת המיקרו-מחשב.

רוב המיקרו-מחשבים הם בעלי מבנה סטנדרטי ולמרות זאת הם יכולים לבצע מספר רב מאד של יישומים שונים ומגוונים. ה-CPU הוא רכיב המסוגל לבצע שורה ארוכה של פעולות, אבל סדר הפעולות שהוא יבצע נקבע בתוכנית העבודה הנמצאת בזכרון המיקרו-מחשב (ב-ROM או ב-RAM).

כדי לבצע את תוכנית העבודה, חייב ה-CPU לעבור (בצורה כלשהי) להוראה הראשונה בתוכנית אותה רוצים שיבצע. משלב זה מתחיל ה-CPU לעבוד בהתאם לתוכנית זו. טיפול בכל הוראה הרשומה בזכרון מחולק לשני שלבים: הבאת ההוראה (באנגלית - Fetch) אל ה-CPU וביצוע ההוראה (באנגלית - Execute). תהליך זה מבוצע ללא הפסק - הבאת הוראה (Fetch) וביצועה (Execute) וחוזר חלילה.

הבאת ההוראה מתבצעת על-ידי הוצאת כתובת התא למסלול הכתובות (הכתובת נמצאת במונה התוכנית) ומסירת אות לזכרון, בעזרת קו מתאים, למסור את תוכן התא שזוהי כתובתו. ההוראה מועברת לאוגר הוראות ב-CPU ומונה התוכנית עולה ב-1. בשלב הביצוע, מפוענחת ההוראה על-ידי יחידת הבקרה שב-CPU ומבוצעת. שלב הביצוע יכול להיות מורכב ממספר מחזורי מכונה בהתאם לסוג ההוראה. בשלב הביצוע עשוי ה-CPU לבצע פעולת קריאה נוספת מתא בזכרון או משני תאים בזכרון או מיחידת I/O כלשהי, באחת משיטות המיעון הידועות. כן עשוי ה-CPU לבצע פעולת כתיבה בתא בזכרון או ביחידת I/O כלשהי, באחת משיטות המיעון.

בחלק גדול של המקרים, מוסר ה-CPU נתון לרכיב שאליו הוא פונה (זכרון או יחידת I/O) ועובר להמשך התוכנית מבלי לוודא שאכן הרכיב קיבל את הנתון, או שה-CPU אוסף נתון ממסלול הנתונים מבלי לוודא קודם שהרכיב אליו פנה אכן מוכן עם הנתון. אם הרכיב איטי מה-CPU, תיתכן הופעת נתונים שגויים. במקרים כאלה עובדים בשיטה הנקראת לחיצת יד (Handshake). בשיטה זו, מקבל ה-CPU (מהרכיב שאליו

במעגלי LSI אלה מאפשר המרת תוכנה בחומרה וקיצור זמן הפיתוח. בפרקים על ממשקים שונים נראה פתרונות בתוכנה והמרתם לפתרונות בחומרה.

1.5 מגבלות המיקרופרוססור

למיקרופרוססור, כרכיב הבנוי על פיסת סיליקון ונתון באריזת DIP (Dual in Line Package), יש מגבלות הנובעות ממספר הפינים האפשרי. מספר הפינים המקובל מוגבל ל-40 עד 42 מסיבות כלכליות. מערכות בדיקה תעשייתיות לבדיקת רכיבים בעלי מספר פינים מעבר ל-42 פינים יקרות מאד בצורה קיצונית. מערכות הבדיקה הסטנדרטיות מקבלות עד 40 או 42 פינים. גם מחיר אריזה של רכיב בעל מספר פינים מעבר ל-40 עולה בצורה דרסטית. למרות זאת ישנם מספר רכיבים ומיקרופרוססורים עם יותר מ-40 פינים. חלק מהמיקרופרוססורים בני 16 ו-32 סיביות באים במארזים שונים ממארזי DIP. ישנם מארזים רבועיים או מארזים בני 4 שורות וכד'.

המגבלה על מספר הפינים באריזת המיקרופרוססור יוצרת את החלוקה הבאה. רוחב סטנדרטי של מסלול הכתובות הוא 16 סיביות, על מנת לאפשר פניה בצורה ישירה ל-64K כתובות אפשריות, ולכן 16 פינים מיועדים לקוי כתובת. 8 פינים מיועדים לקוי נתונים. לפחות 2 פינים מיועדים להספקת מתח לרכיב. 2 פינים למתנד או לחיבור גביש ונשארים 10 עד 12 פינים לקוי בקרה שונים.

במשפחת רכיבי אינטל שבה נעסוק בספר זה, כדי להשיג יותר קוי בקרה, משתמשים ב-8 קוים כדי להעביר את הנתונים ואת שמונת הסיביות הפחות משמעותיות של הכתובת, בזמנים שונים. בתחילה ה-μP רושם בהם את המחצית הנמוכה של

הוא פונה) אות כאשר הוא קלט את המידע או כאשר הוא מוכן עם מידע למסירה. אותות הבקרה השונים ודיאגרמות הזמנים של כל CPU יתוארו בפרקים הבאים.

1.4 תוכנה וחומרה

חלק הארי של זמן הפיתוח של מערכת מיקרו-מחשב מוקדש לתוכנה. חלק מהתוכנה עוסק ביחידות הקלט והפלט וברכיבי הממשק השונים. הפתרון לכל בעיית ממשק ליחידות I/O שונות הוא בשילוב של חומרה ותוכנה. החומרה של רכיבי הממשק יכולה להיות פשוטה, אבל אז נדרשת תוכנה מורכבת לטיפול ביחידת הקלט או הפלט ולהיפך. רכיבי הממשק יכולים להיות חכמים מאד (עד כדי שימוש במיקרו-מחשבים הבנויים על פיסת סיליקון אחת) ואז התוכנה הנדרשת תהיה פשוטה ביותר. נראה זאת בדוגמא הבאה.

נניח שרוצים להוציא תוים למסך. אם המיקרופרוססור עצמו צריך ליצור את האותות הדרושים לכך בזמן המתאים, להמיר מ-ASCII למטריצה המתאימה וכו', אזי בזכרון המיקרו-מחשב צריכה להיות כל התוכנה הדרושה לביצוע הוראות אלה. הוראות אלה צריכות להיות משולבות בתוכנית הראשית של המערכת. לעומת זאת, אם יחידת הממשק תורכב מבקר (Controller) חכם שכל מה שהוא צריך לקבל הוא צופן ASCII והוא ידאג להציג את התו במקום המתאים, אז התוכנה להדפסה במסך תהיה פשוטה ותורכב מהוצאת נתונים בלבד, ליחידת הממשק.

החברות המייצרות רכיבי מיקרו-מחשבים, אכן דואגות לבנות מעגלי LSI ו-VLSI חכמים על מנת להקל על פיתוח תוכנה. חלק ממעגלי LSI אלה הם מיקרו-מחשבים שלמים (כוללים CPU, RAM, ROM ויחידות I/O) הבנויים על פיסה אחת. שימוש

הכתובת ומספק את המאפשר נעילתו ברכיב מיוחד. לאחר מכן מוסר או אוסף נתון דרך קוים אלה. בפרק הבא נראה את דיאגרמות הזמנים ונרחיב את ההסבר על כך.

1.6 מערכת אלקטרונית רגילה מול מערכת מיקרו-מחשב

כפי שתואר בסעיף הראשון של הפרק קיימים הבדלים מהותיים בין מערכת אלקטרונית רגילה לבין מערכת מיקרו-מחשב, אפילו שתיהן ממלאות אותה הפונקציה. חלק מההבדלים הם ההבדלים המתוארים להלן.

המיקרו-מחשב בנוי בצורה סטנדרטית כמעט ללא תלות בפונקציה אותה עליו למלא. הפונקציה שהוא נדרש למלא נקבעת בתוכנה ולכן גם ניתנת לשינוי. במערכת אלקטרונית רגילה, הפונקציה קבועה בחומרה בלבד ולא ניתנת לשינוי. שינוי באיפיון גורר שינוי בבנית המערכת.

במערכת אלקטרונית רגילה זורמים הנתונים מבלוק אחד למשנהו. שינוי נתון כלשהו בכניסה גורם מיד לשינויים באותות הזורמים. ניתן למסור אותות קבועים בכניסה ולעקוב אחרי זרימתם בבלוקים השונים. חיבור המלבנים השונים הוא חיבור רציף. במערכת מיקרו-מחשב, נדגמים הנתונים על-ידי ה-CPU, עוברים אליו, מעובדים שם בעזרת תוכנה הנמצאת בזכרון ומוצאים החוצה. העיבוד הוא עיבוד ספרתי. לא ניתן לעקוב בצורה רגילה אחר זרימת האותות. המערכת היא מערכת סינכרונית ויש להסתנכרן על אות מסוים, כדי לעקוב אחר האות מאותם הנתונים הזורמים במערכת. מרכיבי מערכת מיקרו-מחשב (המלבנים השונים) מחוברים במקביל אל ה-CPU. זרימת האותות השונים היא אל ה-CPU או מה-CPU. שיבוש באחד הקוים גורם, בדרך כלל, להשבתת המערכת כולה. איתור תקלות מחייב בדרך כלל, אמצעים מיוחדים לשם כך.

לעבודה עם מיקרו-מחשב דרושה גישה שונה לגמרי מאשר עבודה עם מערכת אלקטרונית רגילה. המיקרו-מחשב מהווה מהפיכה בדרך הבניה והעבודה של מערכות אלקטרוניות המבוססות עליו. בפרק 2 נתאר שיטות לאיתור תקלות במערכת מיקרו-מחשב.

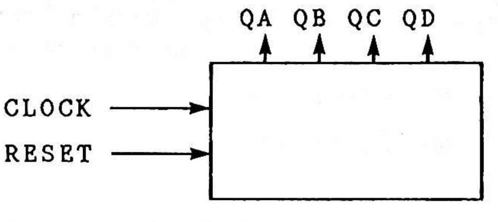
פרק 2

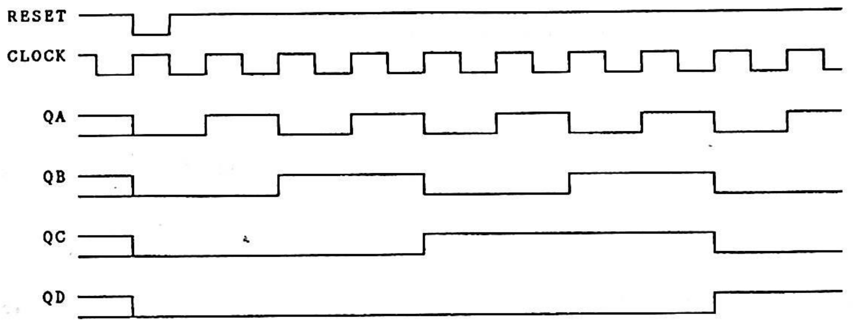
דיאגרמות זמנים ופענוח כתובות

2.1 דיאגרמות זמנים - כללי

דיאגרמת זמנים היא התיאור הפשוט ביותר והברור ביותר שניתן לתאר בו אותות התלויים בזמן. דיאגרמת זמנים מתארת בצורה גרפית מערכת של אותות התלויים זה בזה וכן תלויים בזמן. לא ניתן לתאר מערכת אותות כאלה במערכת צירים רגילה. דיאגרמת זמנים מורכבת ממספר מערכות צירים, כמספר האותות המתוארים, שכל אחת מהן מתארת את אחד האותות בתלות בזמן (לעתים מוותרים על הצירים עצמם ומשאירים זאת לדמיונו של הקורא). צירי הזמן של כל המערכות נמצאים במקביל זה לזה ומתואמים מבחינת הזמנים וקנה המידה. ראשית הצירים של כל המערכות נמצאות זו מתחת לזו. תיאור כזה מאפשר לראות תלות כל אות בזמן ואת התייחסותו לשאר האותות.

לדוגמא, דיאגרמת זמנים של מונה בינארי המתארת את האותות ביציאות המונה ותלותן באותות בשתי הכניסות - כניסת השעון וכניסת ה-RESET.



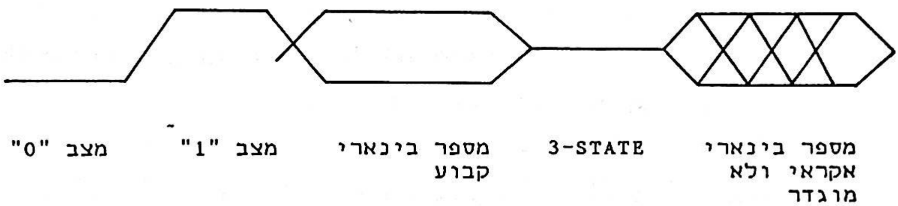


ציור 2-1 דיאגרמת זמנים של מונה בינארי

צורת הסימון, בתחילת תיאור אותות היציאה, מסמנת שאין חשיבות באיזה מצב נמצאות יציאות אלה. כלומר, הן יכולות להיות ב-High או ב-Low. אנו רואים בדיאגרמה המתוארת שהאותות מסונכרנים לזמני עליית השעון (מסמנים זאת בדרך כלל כך ↗).

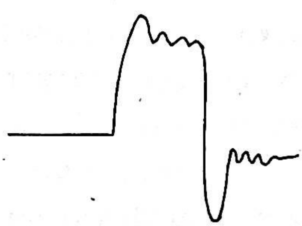
דיאגרמת זמנים של מערכת מיקרו-מחשב מורכבת יותר. בדיאגרמה מתוארים, בנוסף לאותות הבקרה השונים גם האותות במסלולים (במסלול הכתובות ומסלול הנתונים). מסלול הוא אוסף של קוים ויכול להיות במספר מצבים אפשריים. הוא יכול להכיל מספר בינארי כלשהו קבוע בהתאם לתוכנית, להיות במצב ניתוק או להכיל מספרים בינאריים אקראיים ולא מוגדרים (כאשר ה-CPU אינו משתמש ב-BUS).

המצבים האפשריים מתוארים בצורות הבאות:



ציור 2-2 מצבים אפשריים של מסלול

מערכת מיקרו-מחשב היא מערכת מהירה מאד. פולסי השעון הם בתדר גבוה למדי. במערכת כזו קיימת בעיה של התייצבות הפולסים. הפולסים רחוקים מלהיות רבועיים או מלבניים, ולזמני המעבר יש משמעות רבה. בדרך כלל, עקב העליה המהירה של הפולסים מתקבלים דפקים בעלי עליות יתר (Over Shoot), ירידות יתר (Under Shoot) וריסון נמוך. זו הסיבה שמערכת מיקרו-מחשב מבססת את פעולותיה על דגימות של אותות. מתכנני המיקרופרוססורים ורכיבי התמיכה מקדישים מחשבה רבה לזמני התייצבות הפולסים השונים כפי שנראה בהמשך. אות מקובל במיקרו-מחשבים הוא אות מהצורה הבאה:

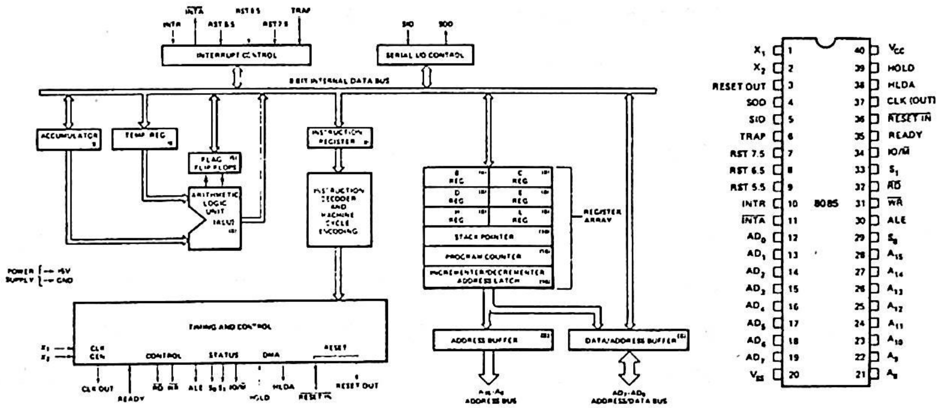


בשיקולי התכנון יש לקחת בחשבון גם את זמן הפענוח. לרכיבי התמיכה לוקח זמן לפענח את הכתובת המצוינת במסלול הכתובות, להעביר וליצב את הנתון מהתא שזוהי כתובתו למסלול הנתונים או להכין את התא המתאים לקליטת הנתון.

מבין שלושת הסעיפים הבאים (2.2, 2.3, 2.4) בחר את הסעיף המתאים למיקרופרוססור איתו אתה עובד.

2.2 רגלי ה-8085 ודיאגרמות הזמנים שלו

ה-8085 הוא µP בן 40 רגליים המתואר בציור הבא:



ציור 3-2 ה-8085 ודיאגרמת המלבנים שלו

תיאור תפקידי הפינים השונים של ה-8085:

<u>x1,x2</u> (פינים 1 ו-2) - חיבורי הגביש, המשמש לקביעת קצב הפעולה של ה-CPU. ל-8085 יש מתנד פנימי שתדירותו נקבעת על-ידי גביש חיצוני.

<u>RESET OUT</u> (פין 3) - זהו קו יציאה מה-CPU, המצביע על כך שמאפסים את ה-CPU.

<u>SOD</u> (Serial Output Data - פין 4) - קו יציאת נתונים טוריים. הוצאת הנתונים הטוריים נעשית בעזרת הפקודה SIM[1].

<u>SID</u> (Serial Input Data - פין 5) - קו כניסת נתונים טוריים. קבלת נתונים טוריים נעשית בעזרת הפקודה RIM[1].

1 מוסבר במפורט בספר "יסודות המיקרו-מחשב 6800 ו-8085" בפרק 10.

__TRAP__ (פין 6) - קו פסיקה לא מסיכה (Non Maskable Interrupt). פסיקה זו היא בעלת העדיפות הגבוהה ביותר. הפסיקה מתקבלת כאשר הקו עולה ל-High ונדגם על-ידי ה-CPU (כלומר ה-CPU מסיים עם ההוראה האחרונה ופונה אליו). פסיקה זו מפנה את ה-CPU לכתובת 0024.

__RST 7.5__ (פין 7) - פסיקה מסיכה בעלת עדיפות מספר 2. הפסיקה מתקבלת כאשר הקו עולה ל-High. הקו איננו חייב להישאר ב-High בזמן דגימת ה-CPU מכיוון שהפסיקה נרשמת ב-CPU (latched) ב-.F.F. פסיקה זו מפנה את ה-CPU לכתובת 003C.

__RST 6.5__ (פין 8) - פסיקה מסיכה בעלת עדיפות מספר 3. הפסיקה מתקבלת כאשר הקו עולה ל-High ונשאר עד שה-CPU דוגם אותו. פסיקה זו מפנה את ה-CPU לכתובת 0034.

__RST 5.5__ (פין 9) - פסיקה מסיכה בעלת עדיפות מספר 4. הפסיקה מתקבלת כאשר הקו עולה ל-High ונשאר ב-High עד שה-CPU דוגם אותו. פסיקה זו מפנה את ה-CPU לכתובת 002C.

__INTR__ (INTerrupt Request - בקשת פסיקה - פין 10) - פסיקה מסיכה בעלת עדיפות מספר 5. הפסיקה מתקבלת כאשר הקו עולה ל-High ונשאר ב-High עד שה-CPU דוגם אותו. לפסיקה זו יש קו נוסף 'INTA ובו ה-CPU מאשר את בקשת הפסיקה ומבקש הוראת הסתעפות.

__'INTA__ (INTerrupt Acknowledge - פין 11) - בקו זה, כאשר הוא יורד ל-Low, מאשר ה-CPU את קבלת בקשת הפסיקה (INTR) ובקש הוראת דילוג. המערכת החיצונית שיצרה את בקשת הפסיקה, צריכה למסור את הוראת הדילוג לקוי הנתונים.

__AD0-AD7__ (Address Data bus - פינים 12-19) - מסלול מעורב של כתובות (8 סיביות תחתונות) ונתונים (8 סיביות). יציאות אלה מסוגלות לעבור למצב 3-State. בחלק הראשון של מחזור המכונה, מופיעות 8 הסיביות של הכתובת ובחלק השני הנתונים.

__VSS__ (פין 20) - קו זה מחובר לקו האדמה של המערכת, המחובר להדק השלילי של ספק כח בן 5V.

A8-A15 (פינים 21-28) - 8 הסיביות המשמעותיות יותר של מסלול הכתובות. גם יציאות אלה מסוגלות לעבור למצב 3-State.

ALE (Address Latch Enable - פין 30) - יציאה זו מאפשרת להפריד בין 8 הסיביות התחתונות של הכתובת ל-8 הסיביות של הנתונים. בזמן הופעת סיביות הכתובת, יציאה זו נמצאת ב-High ועוברת ל-Low בזמן הופעת סיביות הנתונים. ירידה זו מאפשרת לאגור את סיביות הכתובת ב-Latch.

S0,S1 (פינים 29 ו-33) - שתי יציאות אלה מראות את המצב של מחזור המכונה המתבצע באותו זמן.

IO/M' (פין 34) - יציאה זו מסמנת למי פונה ה-CPU, לזכרון או ליחידת קלט/פלט. היציאות S0, S1 ו-IO/M' כותבות את הטבלה הבאה של מצבי מחזור המכונה.

מצב	S0	S1	IO/M'
Memory Write	1	0	0
Memory Read	0	1	0
I/O Write	1	0	1
I/O Read	0	1	1
Opcode Fetch (הבאת הוראה)	1	1	0
Interrupt Acknowledge (קבלת פסיקה)	1	1	1
Halt (עצירה)	0	0	*
Reset	X	X	*

* = 3-State (Hight impedance).

X = לא מוגדר.

מהטבלה רואים ש-S0 ו-S1 יכולים לשמש לסימון מצב קריאה או כתיבה. הבעיה היא שצריך לאגור את תוכנם ב-Latch.

'WR (WRite - פין 31) - מצב Low בקו זה מסמן שהנתונים שבמסלול הנתונים ירשמו ביחידה אליה הם מופנים (בעזרת מסלול הכתובות וקו 'IO/M). יציאה זו עוברת למצב 3-State בזמן עצירת ה-CPU.

'RD (Read - פין 32) - מצב Low בקו זה מסמן ליחידה שאליה פונה ה-CPU להוציא את הנתון למסלול הנתונים. גם יציאה זו עוברת למצב 3-State בזמן עצירת ה-CPU.

READY (פין 35) - כניסה זו מאפשרת לעכב את ה-CPU, כאשר משתמשים בזכרונות איטיים. כאשר הקו נמצא ב-Low, ה-CPU ממתין מבלי להשלים את מחזור הכתיבה והקריאה.

RESET IN (פין 36) - מצב Low בכניסה זו מאפס את מונה התוכנית (PC). קוי הכתובת והנתונים עוברים למצב 3-State. בזמן הפעלת המערכת (חיבור למתח), יש לדאוג שה-CPU לא יבצע תוכנית אקראית כלשהי, אלא תוכנית מסוימת. דבר זה נעשה על-ידי השהיית קו RESET IN ב-Low לאחר שמתח הספק עלה ל-5V. פעולה זו גורמת שה-CPU יתחיל תמיד מכתובת 0000.

CLK (CLocK - פין 37) - זוהי יציאת השעון המאפשרת סינכרון המערכת. זמן המחזור שלה כפול מזמן x1 ו-x2.

HOLD (פין 39) - כניסה זו עוצרת את ה-CPU. עם עליית קו זה ל-High, משלים ה-CPU את השימוש בקוי הכתובת והנתונים ומנתק אותם. פעולתו הפנימית יכולה להימשך. קו זה מאפשר למערכת חיצונית כלשהי לעצור את ה-CPU ולהשתלט על המסלולים.

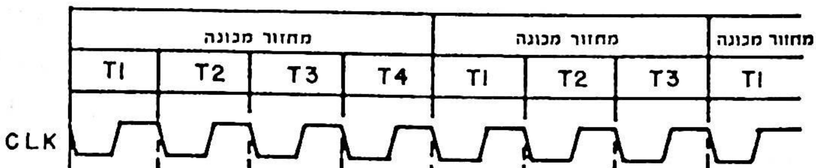
HLDA (HoLD Acknowledge - פין 38) - ביציאה זו מאשר ה-CPU קבלת בקשת Hold וסיום השימוש במסלולים. בזמן HLDA קוי הכתובת, הנתונים, 'RD, 'WR ו-'IO/M עוברים למצב 3-State.

VCC (פין 40) - זהו קו המחובר להדק החיובי של ספק הכח בן 5V.

ה-CPU הוא מערכת סינכרונית, המשנה את מצבה בזמני מעבר של דפקי שעון. שעון ה-8085 יוצר גל ריבועי יחיד. תדר מקובל של השעון הוא 3MHz. ליצירת תדירות זו משתמשים בגביש של 6MHz, כי השעון מחלק ב- 2 את התדירות. ל-8085 יש יציאת אות ברמת TTL המשמשת לסינכרון רכיבי התמיכה של מהיקרופרוססור אליו. יציאה זו היא הקו CLK (CLocK - פין 37).

ביצוע הוראה מתוזמן על-ידי שורה של מחזורי מכונה, שכל אחד מורכב ממספר מחזורי שעון. ב-8085 קיים הבדל בין מחזור מכונה למחזור שעון. מספר מחזורי השעון במחזור מכונה משתנה בהתאם לסוג מחזור המכונה ולסוג ההוראה.

מחזור המכונה הראשון של כל הוראה, יהיה בעל ארבעה או ששה מחזורי שעון. שאר מחזורי המכונה יהיו בעלי שלושה מחזורי שעון. הסיבה לכך היא, שמחזור המכונה הראשון מבצע את הבאת ההוראה (Fetch Instruction). במחזור זה יחידת הבקרה צריכה לפענח את ההוראה המובאת ולבצע אותה אם היא לא כוללת פניה נוספת לזכרון. שאר מחזורי המכונה פשוטים יותר כפי שנראה בהמשך. דוגמא כללית מופיעה בציור 2-3. את מחזורי השעון מציינת האות T (T1,T2,...).



ציור 2-4 אותות השעון במחזורי מכונה של ה-8085

על אף האורך המשתנה של מחזורי המכונה, אין ל-8085 יציאה מיוחדת המשמשת לזיהוי התחלת מחזור מכונה חדש, אך ניתן להשתמש ביציאת ה-ALE למטרה זו. קו ה-ALE (Address Latch

Eanble) עולה ל-High בכל מחזור שעון ראשון של כל מחזור מכונה.

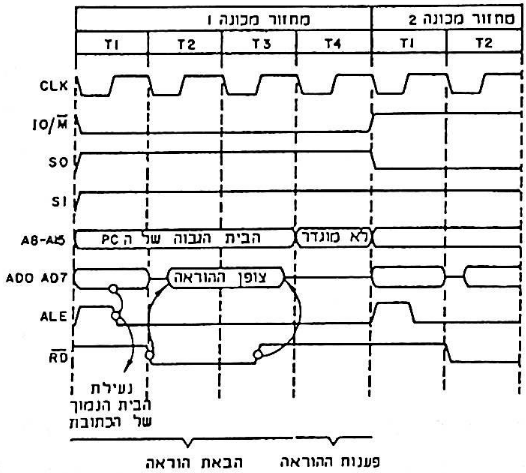
קוי AD0-AD7 משמשים הן להעברת כתובת והן להעברת נתונים בזמני שעון שונים של מחזור המכונה. במחזור השעון הראשון של מחזור המכונה, נמסרות בקוים אלה 8 הסיביות הנמוכות (A0-A7) של הכתובת הדרושה. באותו זמן מופיע דופק High בקו ALE למשך חצי מחזור שעון. דופק זה נועל את 8 הסיביות הנמוכות של הכתובת ברכיב חיצוני, הכולל תאי Latch והמיועד לכך (8212 או 74LS373 למשל).

בשני מחזורי השעון הבאים, מוציא ה-CPU נתון או מקבל נתון על אותם קוים (AD0-AD7). הקוים A8-A15 מוציאים, בכל שלושת מחזורי השעון הראשונים, את 8 הסיביות הגבוהות של הכתובת. מכאן רואים, שלצורך קבלת נתון או מסירת נתון על-ידי ה-CPU, דרושים 3 מחזורי שעון.

בכל מחזור מכונה ב-8085 (מלבד במחזורים של BUS פנוי) קיימת הבאת נתון או מסירת נתון. לכן המספר המינימלי של מחזורי השעון במחזור מכונה הוא 3.

ל-8085 מספר סוגים רב יחסית של מחזורי מכונה. נתאר את אלה הקשורים בזכרון.

2.2.1 מחזור מכונה I - הבאת הוראה (Fetch Instruction):



ציור 2-5 דיאגרמת זמנים של מחזור הבאת הוראה ב-8085

בדיאגרמת זמנים זו יש לשים לב למספר פרטים:

א. השלב שבו כועל קו ALE את הבית (Byte) הכמוך של הכתובת, הוא בירידה שלו ל-Low. משך השהייה שלו ב-High מאפשר לקוי הכתובת להתייצב.

ב. הכתובת במחזור מכונה זה של הבאת ההוראה מגיעה ממונה התוכנית (PC).

ג. קו 'IO/M הוא ב-Low. כלומר, הפניה היא אל הזכרון לצורך הבאת צופן ההוראה (Object Code).

ד. קוים S0 ו-S1, המשמשים לזיהוי סוג מחזור המכונה, נמצאים ב-High, כדי לסמן מחזור קריאה מהזכרון.

ה. קו 'RD נמצא ב-Low בזמן העברת הנתון בלבד. קליטת הנתון על-ידי ה-CPU נעשית עם עליית קו 'RD ל-High.

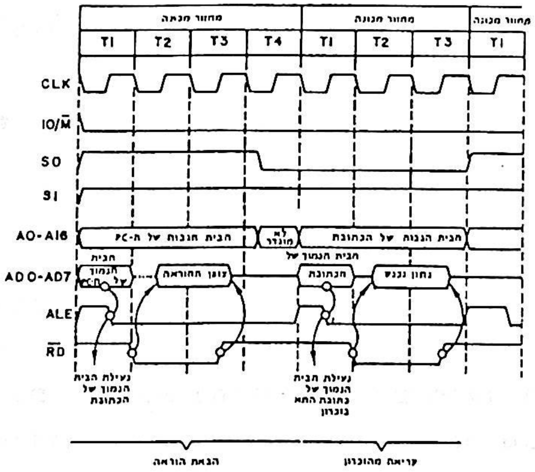
שאר מחזורי השעון של מחזור המכונה הראשון (אחד או שלושה) משמשים את ה-CPU לפענוח ההוראה ולביצועה. בציור מתואר מחזור מכונה ראשון בן 4 מחזורי שעון. אם ההוראה המובאת איכה דורשת פניה נוספת לזכרון (כמו במיעון

מובלע), אזי במחזור המכונה יהיו ששה מחזורי שעון. המחזורים החמישי והששי מיועדים לביצוע ההוראה. בזמן מחזורי השעון הרביעי עד הששי, מנותקים הקוים AD0-AD7 והקוים A8-A15 לא מוגדרים. נקודה זו צריכה להילקח בחשבון בזמן פענוח הכתובת.

אם ההוראה דורשת פניה נוספת לזכרון, יהיה מחזור המכונה הראשון בן ארבעה מחזורי שעון.

2.2.2 מחזור קריאה מהזכרון (Read Machine Cycle):

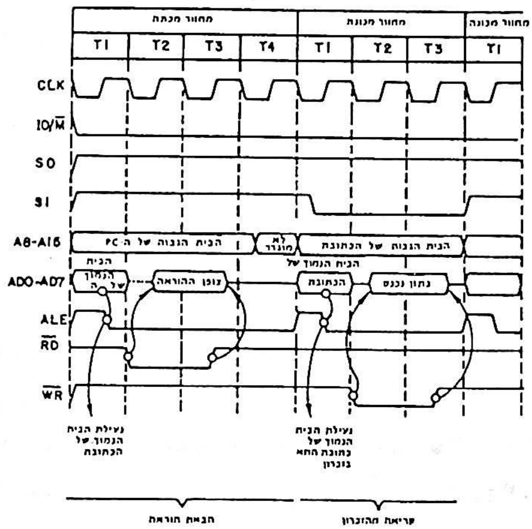
בציור הבא מתוארת דיאגרמת הזמנים של קריאת כתון מהזכרון, הבאה לאחר מחזור מכונה ראשון של הבאת ההוראה. MC1 (מחזור מכונה ראשון - Machine Cycle 1) הוא בן 4 מחזורי שעון, מכיון שההוראה כוללת מחזור מכונה נוסף. מחזור מכונה זה (MC2) הוא בן שלושה מחזורי שעון והוא דומה לשלושת מחזורי השעון הראשונים של MC1, מלבד קוים S0 ו-S1. S0 נמצא ב-Low ו-S1 נמצא ב-High.



ציור 2-6 דיאגרמת זמנים של מחזור קריאה מהזכרון ב-8085

2.2.3 מחזור כתיבה בזכרון (Write Machine Cycle):

מחזור MC2 המתואר בציור הבא, מבצע כתיבה בזכרון לאחר הבאת ההוראה ב-MC1. במחזור זה קו 'WR יורד ל-Low לצורך ביצוע הכתיבה. קו S0 נמצא ב-High וקו S1 ב-Low. קו 'IO/M נמצא ב-Low כדי לציין פניה לזכרון.



ציור 2-7 דיאגרמת זמנים של מחזור כתיבה בזכרון ב-8085

ל-8085 יש סוגים נוספים של מחזורי מכונה אבל לא נדון בהם פה. ניתן למוצאם בדפי הנתונים של הרכיב.

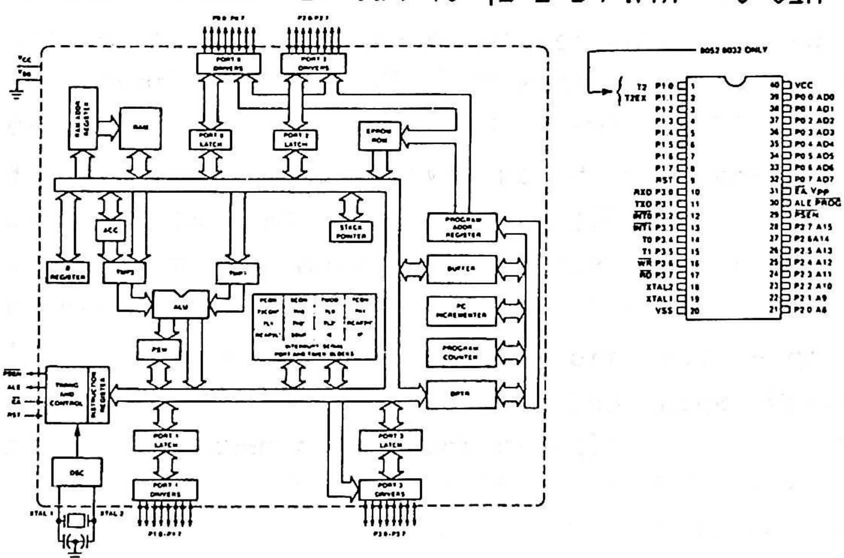
נזכיר רק שמחזורי פניה ליחידות I/O דומים למחזורי קריאה וכתיבה לזכרון מלבד העובדה שקו 'IO/M נמצא ב-High. גם קוים S0 ו-S1 שונים. מעגלי הפענוח אינם מתייחסים לקוים S0 ו-S1. גם אנו לא נתייחס אליהם מכאן ואילך. קוים אלה משמשים יחידות שחשוב להם לדעת באיזה מצב נמצא ה-CPU. לדוגמא, האם הקריאה מהזכרון היא לשם הבאת הוראה או קריאת נתון בלבד.

2.3 רגלי ה-8051 ודיאגרמות הזמנים שלו

כמה דברים מיוחדים ברכיב 8051. ראשית זהו רכיב בקר זעיר (µC - Microcontroller), השייך למשפחת הרכיבים המכונים מיקרו-מחשב בפיסה אחת (One Chip Microcomputer). מיקרו-מחשב שלם הכולל CPU, ROM, RAM ויחידות I/O שונות, הנמצאים כולם במעגל משולב אחד. מצד שני רכיב זה יכול לתפקד כ-CPU הניתן להרחבה בעזרת רכיבי תמיכה. לרגליו יש תפקוד נוסף לשם העברת קוי כתובת, נתונים ובקרה.

דבר נוסף לגבי ה-8051. זוהי משפחה עניפה הכוללת רכיבים רבים הניתנים לשימוש במיגוון רחב של יישומים. לכולם גרעין זהה ואוצר פקודות אחיד. מעבר לגרעין יש דברים המיוחדים כל µC במשפחה והמתאימים אותו ליישומים מסוימים.

בספר זה נתייחס לתפקוד ה-8051 כ-CPU. נלמד כיצד להרחיבו מבחינת זכרון ולחבר אליו רכיבי תמיכה שונים. ה-8051 הבסיסי הוא רכיב בן 40 רגליים המתואר בתמונה הבאה.



ציור 2-7 תיאור ה-8051 ורגליו

אופן ההפעלה והשימוש במרכיבים הפנימיים של ה-8051 , כמו הפורטים, המונים/קוצבי זמן, ה-UART (רכיב התקשורת הטורית) והפסיקות מתוארים בספר "יסודות הבקר הזעיר 8051" בהוצאת SES.

VCC (פין 40) - מתחבר להדק החיובי של ספק הכח בן 5V±10%.

VSS (פין 20) - קו האדמה של המערכת. מתחבר להדק השלילי של ספק הכח.

XTAL1 ,XTAL2 (פינים 18 ו-19) - רגלי חיבור לגביש. Clock ה-8051 הוא פנימי והוא נעזר בגביש חיצוני. ניתן לחבר גם Clock חיצוני ואת יציאתו לחבר ל-XTAL1 או ל-XTAL2 בהתאם לסוג ה-8051 (ראה דפי נתונים של הרכיב).

RST (פין 9) - כניסת היתחול (RESET). עם עלייתה ל-High ה-CPU מפסיק את פעולתו. עם ירידתה ל-Low ה-CPU מבצע פעולות היתחול ומתחיל בתוכנית החל מכתובת 0000. פעולות ההיתחול מתוארות בספר "יסודות הבקר הזעיר 8051".

EA (External Address - פין 31) - כניסה זו מסמנת ל-CPU האם איזור ה-Program בתחום הכתובות הנמוך (0000-0FFF ב-8051), שייך ל-ROM חיצוני או ל-ROM הפנימי שבתוכו. בכל מקרה של שימוש ב-ROM חיצוני לתחום כתובות זה (גם ברכיב כמו ה-8031 שאינו כולל ROM פנימי), יש לקצר קו זה ל-GND.

ALE (פין 30) - קו יציאה הנועל את הכתובת הנמוכה ב-Latch (Address Latch Enable). עולה ל-High למשך state אחד (2 מחזורי שעון) בכל פעם שמתבצעת פניה לזכרון.

PSEN' (Program Set ENable - פין 29) - קו יציאה, המציין לזכרונות ה-Program שה-CPU מבקש לקרוא נתון מאיזור זה, כאשר קו זה יורד ל-Low.

P0.0-P0.7/AD0-AD7 (פינים 32-39) - קוי פורט 0, היכולים לשמש כפורט מבוא או כפורט מוצא. במקרה של חיבור רכיבי תמיכה חיצוניים לרכיב, מתפקדים קוים אלה כקוי AD0-AD7, ומעבירים את החלק הנמוך של הכתובת בשלב הראשון של מחזור המכונה ואת הנתונים בשלב השני של מחזור המכונה.

P1.0-P1.7 (פינים 1-8) - קוי פורט 1, היכולים לשמש כפורט מבוא או כפורט מוצא.

P2.0-P2.7/A8-A15 (פינים 21-28) - קוי פורט 2, היכולים לשמש כפורט מבוא או כפורט מוצא. במקרה של חיבור רכיבי תמיכה חיצוניים לרכיב, מתפקדים קוים אלה כקוי הכתובת הגבוהים A8-A15.

P3.0-P3.7 (פינים 10-17) - קוי פורט 3, היכולים לשמש כפורט מבוא או כפורט מוצא. בנוסף לכך יכולים קוים אלה לתפקד כקוי בקרה שונים, המתוארים להלן.

P3.0/RXD (פין 10) - קו כניסה של אות המגיע בתקשורת טורית אסינכרונית ל-UART.

P3.1/TXD (פין 11) - קו יציאה של אות היוצא בתקשורת טורית אסינכרונית מה-UART.

P3.2/INT0' (פין 12) - קו פסיקה חיצונית היכול לעבוד בדרבון רמה או בדרבון קצה (ראה ספר "יסודות הבקר הזעיר 8051").

P3.3/INT1' (פין 13) - קו פסיקה חיצונית היכול לעבוד בדרבון רמה או בדרבון קצה (ראה ספר "יסודות הבקר הזעיר 8051").

P3.4/T0 (פין 14) - כניסת אותות ל-Timer0 במקרה שהוא מתפקד כמונה.

P3.5/T1 (פין 15) - כניסת אותות ל-Timer1 במקרה שהוא מתפקד כמונה.

P3.6/WR' (פין 16) - קו יציאה המציין ליחידות RAM חיצוני, שה-CPU מבקש לכתוב נתון.

P3.7/RD' (פין 17) - קו יציאה המציין ליחידות RAM חיצוני, שה-CPU מבקש לקרוא נתון.

לגבי התפקוד הכפול והמשולש של הפורטים צריך לציין מספר נקודות.

א. סיביות הפורטים ניתנות לפניה במיעון סיבית, כלומר, לכל סיבית בנפרד. כל סיבית יכולה לשמש כפורט מבוא או כפורט מוצא בלי קשר לאחרות.

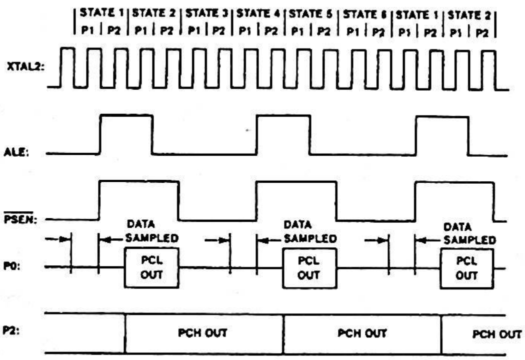
ב. כאשר סיבית מיועדת לתפקד כפורט מבוא, יש להוציא לסיבית המוצא "1", מכיוון שיציאת F.F. סיבית המוצא וכניסת סיבית המבוא קשורות ביחד לרגל הרכיב. אם יהיה "0" ביציאת ה-.F.F, זה יאלץ "0" בקו בלי קשר לאות המגיע מהעולם החיצון.

ג. גם כאשר מיועדת הסיבית לתפקד כקו כתובת או כקו AD או כקו בקרה, יש להוציא "1" לסיביות המוצא, מהסיבה הרשומה בסעיף ב'. זו הסיבה, שבהיתחול ה-8051 (ב-Power Up או לאחר RESET), מקבלים כל הפורטים את הערך FF.

כאמור ה-CPU היא מערכת סינכרונית, המשנה את מצבה בהתאם לדפקי שעון. השעון ב- 8051 נמצא בתוכו, ונעזר בגביש חיצוני הקובע את תדר התנודות. ב-8051 מקובל להשתמש בגבישים בתחום שבין 6MHz ועד 15MHz. גביש נוח במיוחד הוא גביש בעל תדר של 11.0592MHz, המאפשר עבודה מדויקת בקצבי שידור וקליטה (baud rate) מקובלים (ראה ספר "יסודות הבקר הזעיר 8051").

ל-8051 אין יציאת Clock. רכיבי התמיכה מסתנכרנים אל קוי ה-'RD, ה-'WR, ה-'PSEN וה-ALE ואלה מסונכרנים לשעון. אם רכיב תמיכה זקוק ל-Clock, המסונכרן ל-Clock ה-CPU עצמו, מבצעים את הפעולה הבאה. בונים Clock חיצוני ואת יציאתו מספקים לאחת מכניסות הגביש (הכניסה משתנה בהתאם לטכנולוגית ה-μP ויש להתבונן בדפי הנתונים) ואל רכיב התמיכה.

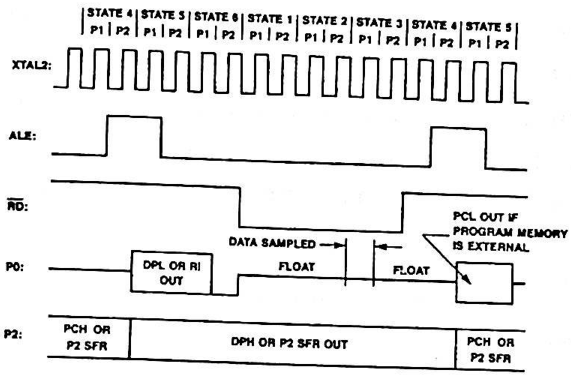
מחזור מכונה של ה-8051 מורכב מ-12 מחזורי שעון המחולקים ל-6 states. כל state מורכב משני מחזורי שעון P1 ו-P2. בכל מחזור מכונה אחד פונה ה-CPU לזכרונות פעמיים, בדרך כלל, כפי שנראה בציור הבא, המתאר פניה של ה-CPU לאיזור Program לשם הבאת OP. Codes (Inst. In בציור).



ציור 2-8 דיאגרמת זמנים בפניה לאיזור ה-Program

איזור ה-Program הוא ROM, בדרך כלל, המכיל את תוכנות המחשב. ה-CPU פונה אליו אוטומאטית, כאשר הכתובת שהוא מוציא היא כתובת הנמצאת במונה התוכנית (Program Counter). תיתכן גם פניה באמצעות הפקודה MOVC, הטוענת את הצובר בנתון מאיזור ה-Program.

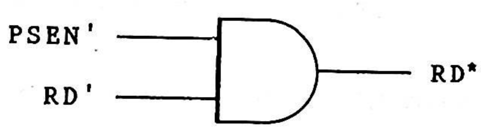
ה-8051 יכול לכלול גם זכרונות הנקראים RAM חיצוני (External RAM). הפניה לזכרונות אלה נעשית בעזרת הפקודה MOVX בלבד, לקריאה או לכתיבה. בחומרה נעשית הפניה בעזרת קוים מיוחדים 'RD ו-'WR הנפרדים לחלוטין מקו 'PSEN. כאשר מזהה ה-CPU פקודה הכוללת קריאה מה-RAM החיצוני באמצעות הפקודה MOVX, מתבצעת דיאגרמת הזמנים הבאה.



ציור 2-9 דיאגרמת זמנים בקריאה מ-RAM חיצוני

בכתיבה ל-RAM חיצוני, הנעשית גם כן בעזרת הפקודה MOVX, דיאגרמת הזמנים דומה לציור הקודם מלבד שני הבדלים. במקום קו 'RD היורד ל-"0", יורד ל-"0" קו 'WR. הנתון הנרשם בקוי הכתובים בזמן ירידת קו 'WR ל-"0", הוא נתון המגיע מה-CPU והנקרא Data Out במקום Data In.

יתכן מצב שנרצה לאחד את איזורי הזכרון. כלומר, ליצור איזור זכרון, שהפניה אליו תהיה גם באמצעות קו 'PSEN וגם באמצעות קו 'RD. במקרה כזה יוצרים קו משותף מפעולת AND על שני הקוים.



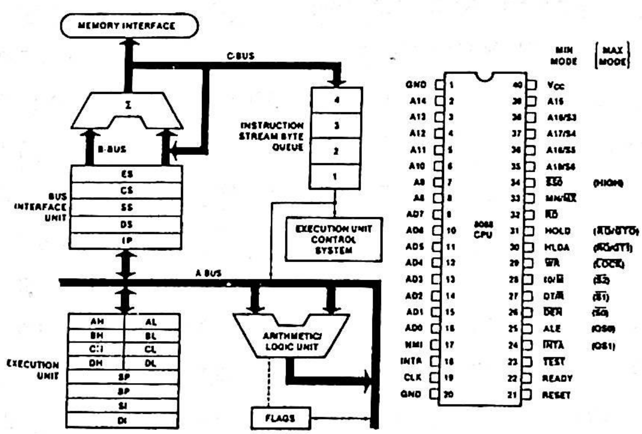
קו *RD ירד ל-"0" כאשר קו 'PSEN ירד ל-"0" או כאשר קו 'RD ירד ל-"0".

2.4 רגלי ה-8088 ודיאגרמות הזמנים שלו

מבחינת חומרת ה-8086/8088 עוסק ספר זה ב-8088. ה-8088 הוא 8086 במארז 8 סיביות. כל אוצר הפקודות זהה לחלוטין. גם רוב הקוים דומים. כל ההבדל הוא שיצרו ב-8088 אופן פניה לרכיבי התמיכה בדומה למיקרופרוססורים בני 8 סיביות למרות היותו מיקרופרוססור בן 16 סיביות. בצורה כזו ניתן לחבר אליו בפשטות, זכרונות ורכיבי I/O שונים, הבנויים למסלולי נתונים בני 8 סיביות.

גם מבחינת המחשב המארח אין זה משנה. אנו מתרגלים בעזרת מחשב IBM PC או תואם, שבתוכו נמצא כרטיס APPLIC-1. לכרטיס זה יציאת BUS בת 8 קוי נתונים בלי קשר אם יש בתוכו 8086 או 8088. את תרגילי הרחבת הזכרון לא נוכל לבצע באמצעות APPLIC-1 עקב מגבלות המערכת, אבל את כל התרגילים הקשורים ביחידות הקלט/פלט השונים נוכל לבצע.

ה-8088 הוא µP בן 40 רגליים, המתואר בציור הבא:



ציור 2-10 תיאור ה-8088 ורגליו

ל-8088 כמו ל-8086 יש שני אופני עבודה. Minimum Mode ו-Maximum Mode. יש רגל הקובעת באיזה אופן עבודה אנו משתמשים. אנו נתאר את תפקידי רגליו ב-Minimum Mode וזה האופן שנתייחס אליו בכל הספר בהמשך. ההבדל בין שני האופנים מתואר בתיאור הרגל 'MN/MX המופיע בהמשך.

<u>VCC</u> (פין 40) - מתחבר להדק החיובי של ספק הכח בן 5V±10%.

<u>GND</u> (פינים 1 ו-20) - קו האדמה של המערכת. מגיע מההדק השלילי של ספק הכח.

<u>S6-S6/A19-A16</u> (פינים 35-38) - קוי הכתובת הגבוהים ביותר. בפניה לכתובת I/O, קוים אלה במצב "0". במחזורי השעון T2,T3,T4 קוים אלה מציינים את אינפורמצית המצב (Status) של מחזור המכונה. מידע כמו האם זה צופן הוראה או נתון, האם מידע למחסנית, האם מצב פסיקה מאופשר או לא. כל זאת לטובת מערכות, היודעות לפענח קוים אלה.

<u>A15-A8</u> (פינים 2-8 ו-39) - קוי הכתובת הגבוהים A15-A8. לאורך כל מחזור המכונה קוים אלה לא משתנים ואין צורך לנעול אותם באמצעות Latch.

<u>AD7-AD0</u> (פינים 9-16) - קוי כתובת/נתונים המציינים בתחילת מחזור המכונה, כאשר קו ALE ב-High, את הכתובת ובהמשך מחזור המכונה את הנתון שמוציא ה-CPU, או מקבלים אל ה-CPU את הנתון שנמסר ל-CPU.

<u>'RD</u> (פין 32) - קו המציין לרכיבי התמיכה שה-CPU מבקש לקרוא נתון כאשר קו זה יורד ל-Low. הקו מסונכרן לשעון ויורד ל-Low במהלך T2 ו-T3.

<u>READY</u> (פין 22) - קו זה מגיע מרכיב השעון 8284, הכולל בכניסתו קו RDY. קוים אלה משמשים להארכת מצב 'RD, כאשר מחברים רכיבים איטיים למערכת. במקרה כזה, מוסר הרכיב את RDY ל-8284 והאחרון מוריד ל-Low את קו READY, המסמן ל-CPU שרכיב התמיכה לא מוכן למסירת הנתון. אז מוסיף ה-CPU מחזור שעון נוסף ומאריך את זמן ה-'RD.

<u>INTR</u> (פין 18) - קו כניסה המגיע מבקר פסיקות מיוחד,

המספק לאחר בקשת הפסיקה את כתובת וקטור הפסיקה, בהתאם לבקשה שהתקבלה אצלו. ה-CPU מדבר עם בקר הפסיקות בעזרת קו INTA, שמתואר בהמשך.

NMI (פין 17) - כניסת בקשת פסיקה לא מסיכה (Non Maskable Interrupt). זוהי פסיקה שלא ניתנת לחסימה, הנוצרת במעבר קו ה-NMI מ-Low ל-High.

TEST (פין 23) - ל-8088/8086 יש פקודה הנקראת "Wait for test". כאשר מתקבלת פקודה כזו, עוצר ה-CPU בכל פעם שקו TEST נמצא ב-High. הורדתו ל-Low גורמת להמשך הביצוע. ניתן לעצור את ה-CPU בכל מחזור שעון.

RESET (פין 21) - קו כניסה המגיע מרכיב השעון 8284. עלייתו ל-High גורמת לעצירת ה-CPU ולהיתחולו מחדש. קו RESET חייב להיות ב-High לפחות ארבעה מחזורי שעון.

CLK (פין 19) - קו כניסה המגיע מרכיב השעון 8284. זו אות של שעון המערכת אליו מסונכרנים כל האותות והרכיבים.

MN/'MX (פין 33) - קו כניסה המסמן ל-CPU אם הוא עובד ב-Minimum Mode או ב-Maximum Mode. כל ההבדל בין שתי שיטות אלה הוא בשיטת הפניה לרכיבי התמיכה. ב-Minimum Mode שיטת הפניה היא כמו ב-8085 בעזרת הקוים 'IO/M, 'RD ו-'WR, המסמנים אם הפניה היא ליחידת קלט/פלט או זכרון לצרכי קריאה או כתיבה. ב-Maximum Mode משתמשים בארבעה קוי בקרה הנקראים 'MRDC (Memory Read), 'MWRC (Memory Write), 'IORDC (IO Read) ו-'IOWRC (IO Write). קוים אלה בנוסף לקוים 'DEN, 'DT/R, ALE, 'INTA, נוצרים על-ידי רכיב נוסף המתחבר ל-8088 והנקרא 8288 (Bus Controller). רכיב זה מפענח את קוי המצב S0,S1,S2 הנוצרים במקרה של עבודה ב-Maximum Mode.

בשיטת Maximum Mode התפנו מספר קוים נוספים ב-8088 ואלה יועדו לאפשר לכמה מיקרופרוססורים לעבוד עם אותו BUS, כלומר שבכל פעם יוכל µP אחר להשתלט על רכיבי התמיכה. לא נעסוק בתחום זה בספר זה.

הקוים הבאים ב-CPU קיימים ב-Minimum Mode בלבד.

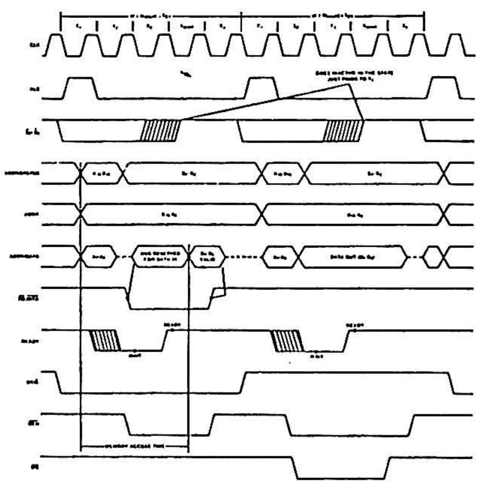
<u>IO/M'</u> (פין 28) - קו יציאה המסמן פניה ליחידות IO במקרה והוא "1" או פניה ליחידת זכרון במקרה והוא "0".
<u>WR'</u> (פין 29) - קו המציין לרכיבי התמיכה שה-CPU מבקש לכתוב נתון. קו זה יורד ל-Low במחזורי השעון T2 ו-T3.

<u>INTA'</u> (פין 24) - קו היורד ל-Low כאשר מתקבלת בקשת פסיקה דרך קו INTR ומשמש כקו READ מבקר הפסיקות.
<u>ALE</u> (Address Latch Enable - פין 25) - קו יציאה הכועל את הכתובת ב-Latch. עולה ל-High בתחילת כל מחזור מכונה למשך מחזור שעון אחד.
<u>DT/R'</u> (Data Transmit/Receive - פין 27) - קו יציאה המסמן לחוצץ נתונים אם כיוון זרימת הנתונים הוא אל ה-CPU או מה-CPU. מוסבר בהרחבה בסעיף 3.7 העוסק בחוצצים.
<u>DEN'</u> (Data ENable - פין 26) - קו יציאה המסמן לחוצץ נתונים אם צריך להעביר נתונים או לא. גם קו זה מוסבר בסעיף 3.7 .
<u>HOLD</u> (פין 31) - קו כניסה שהעלתו ל-High גורמת לעצירת ה-CPU לאחר השלמתו את מחזור המכונה האחרון. כאשר ה-CPU עוצר הוא מסמן זאת ליחידה המבקשת בעזרת קו HLDA ומעביר את קוי המוצא שלו, מלבד קוים 'INTA ו-ALE, למצב 3-State. הורדת קו HOLD משחרר את ה-CPU, הממשיך מהנקודה בה הפסיק.
<u>HLDA</u> (HoLD Acknowledge - פין 30) - קו שבעזרתו מאשר ה-CPU את כניסתו למצב "HOLD" והתנתקותו מקוי ה-BUS.
<u>SSO'</u> (פין 34) - קו Status שניתן ללמוד ממנו ומהקוים DT/R' ו-IO/M' לגבי סוג מחזור המכונה המתבצע לפי הטבלה הבאה:

סוג המחזור	SSO'	DT/R'	IO/M'
Interrupt Acknowledge	0	0	1
Read I/O port	1	0	1
Write I/O port	0	1	1
Halt	1	1	1
Code Access	0	0	0
Read memory	1	0	0
Write memory	0	1	0
Passive	1	1	0

ה-Clock ב-8086/8088 הוא Clock חיצוני ונוצר בעזרת רכיב הנקרא 8284. רכיב זה יוצר את אותות השעון ל-8086/8088 ולרכיבי התמיכה וכן את אות ההיתחול RESET. הוא גם מסמן ל-CPU במידה ואחד מרכיבי התמיכה מבקשים לעצור אותו לזמן מסוים. הגביש מתחבר ל-8284. תדרים מקובלים הם בין 5MHz-10MHz.

מחזור מכונה של ה-8088 מורכב מארבעה מחזורי שעון. בדיאגרמה הבאה מתוארים הקוים העיקריים (שאליהם נתייחס בהמשך הפרקים), המשתתפים בשני מחזורי מכונה. מחזור המכונה הראשון הוא מחזור קריאה מהזכרון והמחזור השני הוא כתיבה בזכרון.



ציור 2-11 דיאגרמת זמנים של ה-8088

לקוים 'DT/R ו-'DEN נתייחס בפרק 3 בסעיף 3.7. פניה ליחידות I/O נעשית בצורה דומה רק קו 'IO/M נמצא ב-High ולא ב-Low.

2.5 סוגי זכרונות

זכרונות מערכת מיקרו-מחשב מחולקים לשלושה סוגים עיקריים:

א. זכרונות קריאה/כתיבה הנקראים RAM (Random Access Memory - זכרון גישה אקראית) שם הנובע מסיבה היסטורית.

ב. זכרונות קריאה בלבד הנקראים ROM (Read Only Memory - זכרון קריאה בלבד).

ג. זכרונות חיצוניים.

הזכרונות החיצוניים אינם מחוברים ישירות אל ה-CPU. הם קשורים למסלולים השונים בעזרת מערכת ממשק מתאימה (Interface). זכרונות אלה ניתנים לקריאה ולכתיבה. הזכרונות הם זכרונות מגנטיים כמו תקליטים (דיסקים), תקליטונים (דיסקטים), סרטים מגנטיים, זכרון בועות (Bubble Memory) ועוד. הזכרונות החיצוניים מאפשרים אגירת תוכניות ונתונים בצורה שלא נמחקת עם כיבוי המתח למערכת ומאפשרים כתיבה מחדש. קיבול זכרון חיצוני הוא בדרך כלל גדול מאד. בעייתם היא גודלם הפיסי והצורך בכוננים וביחידות ממשק כדי להשתמש בהם. גם זמן הגישה אל הנתונים האגורים בהם הוא גדול יחסית ל-ROM ול-RAM.

זכרון קריאה בלבד (ROM) הוא זכרון שבמצב עבודה רגיל של המערכת אין אפשרות לכתוב בו. ניתן לקרוא ממנו נתונים או להריץ תוכניות הרשומות בו, אבל לא לאחסן בו נתונים. תוכן תאיו איננו נמחק עם כיבוי המתח למערכת. הוא אכן משמש לאגירת תוכניות קבועות ונתונים קבועים שאינם משתנים במהלך התוכנית.

מבחינים בסוגים שונים של זכרונות קריאה בלבד:

א. MASK ROM - זהו זכרון קריאה בלבד שתוכן תאיו נקבע במפעל המייצר אותו. מזמין הרכיב מוסר למפעל המייצר רכיבים, את תוכן התאים הדרוש, והאחרון קובע תוכן זה על-ידי הוספת או ביטול חיבורים חשמליים במהלך הייצור. בסוג זה של ROM משתמשים במערכות זהות המיוצרות בכמויות כאלה שישתלם להזמין במפעל ROM מתוכנת. בכמויות גדולות, סוג זה של ROM הוא הזול ביותר.

ב. PROM (ROM - Programmable ROM הניתן לתיכנות) - תוכן התאים של ROM כזה יכול להיקבע על-ידי המזמין בעזרת מערכת מתאימה לא מסובכת במיוחד. ב-PROM המגיע מבית החרושת, כל סיביות התאים נמצאות בדרך כלל ב-High. קביעת תוכנו נעשית על-ידי פניה לתאיו, כתובת אחר כתובת, ומסירת פולס מתח גבוה או פולס זרם גבוה לסיביות הנתונים המתאימות. פולס זה גורם לנתיכים פנימיים להישרף ולירידת אותן סיביות ל-Low (יש רכיבי PROM המגיעים עם סיביות ב-Low ומעלים חלק מהן ל-High). תוכן תאי ה-PROM כמו ב-ROM ניתן להיקבע פעם אחת בלבד. אין אפשרות להחליף את תוכן תאיו.

ג. EPROM (PROM - Erasable PROM הניתן למחיקה) - זכרון ROM זה דומה לזכרון ה-PROM בכך שניתן לתכנת אותו בעזרת מערכת תכנות מתאימה על-ידי המשתמש, בדומה לתכנות ה-PROM. הבדל ביניהם הוא, שתוכן תאי ה-EPROM ניתן למחיקה ובכך גם לתכנות מחדש. הבדל זה נובע מכך שבצריבת ה-EPROM לא ניתכים נתיכים פנימיים אלא פולס המתח הגבוה גורם להצטברות מטענים בשערים פנימיים המופרדים בעזרת תחמוצת הצורן (תחמוצת הצורן היא מבדד מעולה). הפרדה זו גורמת לאי התפרקות המטען ובכך לשמירת המידע עם כיבוי המתח למערכת.

מחיקת ה-EPROM נעשית על-ידי חשיפת תאיו לאור אולטרא

סגול. אור זה גורם לתחמוצת הצורן להוליך זרם במידה מספקת כדי לפרוק את המטענים האגורים והחזרתו למצב הראשוני. זו הסיבה שברכיב ה-EPROM יש חלון שקוף דרכו ניתן להעביר את האור האולטרא סגול. ה-EPROM יקר מה-PROM וה-ROM ומשמש בעיקר לבניית אב טיפוס או במערכות המיוצרות בסדרות קטנות.

ד. EAROM (Electrically Alterable ROM - ROM הניתן לשינוי בצורה חשמלית) - רכיב זה נקרא גם EEPROM (Electrically Erasable PROM - PROM הניתן למחיקה בצורה חשמלית). הוא דומה ל-ERPOM מלבד העובדה שהוא ניתן למחיקה בצורה חשמלית ולא על-ידי אור אולטרא סגול. המחיקה נעשית על-ידי מתן פולס מתח גבוה אחר מאשר בתכנותו (קיימים כבר EAROM הנמחקים על-ידי 5V). היתרון ברכיב כזה, שהוא יכול לשמש לצורך אגירת נתונים ותוכניות במערכת. אפשר להשתמש במקום ROM שניתן להחליף את תוכנו. אי אפשר להשתמש בו במקום RAM כי תהליך הכתיבה איטי ביחס לתהליך הקריאה.

הסבר נוסף נמצא בהמשך.

זכרונות קריאה/כתיבה (RAM) הם זכרונות שזמן הגישה לתא בקריאה שווה לזמן הגישה לתא בכתיבה (להבדיל מה-EAROM). משמעות השם גישה אקראית הוא שזמן הגישה לכל תא ביחידת הזכרון שווה בלי תלות במיקום התא ברכיב, בהבדל מזכרון בעל גישה טורית. גם זכרונות הקריאה בלבד הינם בעלי גישה אקראית, אבל השם RAM נשאר לזכרונות הקריאה/כתיבה. לכן כשאומרים RAM מתכוונים לזכרון קריאה/כתיבה. בכיבוי המתח, המידע האגור ב-RAM נמחק. בחיבור המתח, המידע ב-RAM אקראי.

זכרונות ה-RAM מחולקים לשני סוגים - RAM סטטי ו-RAM דינמי. ה-RAM הסטטי שומר את סיביות המידע בפליפ-פלופים פנימיים ומידע זה נשמר כל זמן שהמתח מחובר למערכת.

ב-RAM הדינמי, נשמרות סיביות המידע באלמנטים קיבוליים. כדי לשמור על המידע יש לרענן אלמנטים אלה ולמנוע התפרקות המטענים במחזוריות מסוימת. שימוש ב-RAM דינמי מחייב חיבור מעגל רענון מיוחד למערכת, דבר המסבך את המערכת במקצת. מצד שני, RAM דינמי הוא בעל נפח זכרון גדול בהרבה מה-RAM הסטטי. RAM דינמי שימושי ביחידות זכרון RAM גדולות מאד.

בין הזכרונות מבדילים גם מבחינת טכנולוגיות הייצור שלהם. טכנולוגיות הייצור זה ספר בפני עצמו, ולא נכנס לכך. נשווה בין התכונות החשובות למשתמש של הטכנולוגיות השונות. התכונות החשובות ביותר הן: מחיר, מהירות, צפיפות רכיבים בפיסה וצריכת הספק.

רכיב זכרון אידיאלי הוא רכיב בעל מחיר נמוך, מהירות גבוהה, צפיפות רכיבים (קיבול זכרון) רבה וצריכת הספק נמוכה. הבחירה ברכיב המתאים נעשית כפשרה בין התכונות השונות ובהתאם לתכונה הדומיננטית הדרושה.

טכנולוגיות הייצור של רכיבי זכרון המקובלות היום הן:

BIPOLAR - שימוש בטרנזיסטורים בי-פולריים בלוגיקת TTL או ECL.

:MOS FET

Positive Channel Metal Oxide Semi Conductor - PMOS
Negative Channel Metal Oxide Semi Conductor - NMOS
Complementary Metal Oxide Semi Conductor - CMOS
Vertical Metal Oxide Semi Conductor - VMOS
Metal Nitride Oxide Semi Conductor - MNOS

השוואת מהירות:

המהיר ביותר - BIPOLAR
VMOS
NMOS
CMOS
PMOS
האיטי ביותר - MNOS

השוואת קיבול זכרון - סיביות לרכיב:

הצפוף ביותר - MNOS
CMOS
VMOS
NMOS
PMOS
הפחות צפוף - BIPOLAR

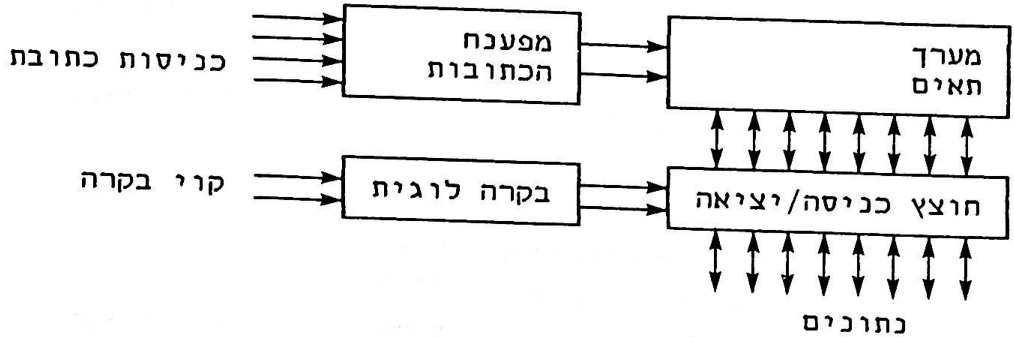
השוואת צריכת אנרגיה:

צריכת הספק נמוכה - MNOS
CMOS
VMOS
NMOS
PMOS
צריכת הספק גבוהה - BIPOLAR

השוואת מחיר:

זול יותר - MNOS
PMOS
NMOS
CMOS
VMOS
יקר יותר - BIPOLAR

בפרק זה אנו דנים בחיבור יחידות זכרון בעלי גישה אקראית והניתנים לתיכנות כמו PROM, EPROM, EAROM ו-RAM אל ה-CPU. יחידות זכרון אלה ערוכות במטריצה של תאים כך שזמן הגישה לכל תא שווה, כמתואר בציור 12-2.



ציור 12-2 תיאור מלבני של יחידת זכרון

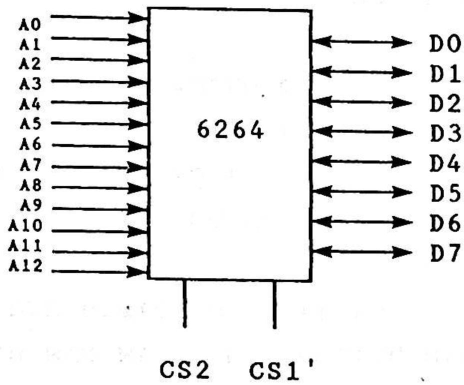
מספר קוי הכתובת הוא בהתאם למספר התאים. ל-n קוים יש 2^n צירופים שונים ולכן הם מאפשרים פניה ל-2^n תאים שונים. לדוגמא, ליחידת זכרון בת 1024 תאים יש 10 קוי כתובת (2^{10} = 1024 = 1K). מספר התאים משתנה מרכיב לרכיב. צירוף בינארי על קוי הכתובת קובע איזה תא מופנה אל חוצץ קוי הנתונים. הפניה יכולה להיות הן לצורך קריאה מהתא והן לצורך כתיבה בתא.

מספר קוי הנתונים הוא בהתאם לגודל התא. קיימות יחידות זכרון בעלות תאים בני סיבית אחת עד יחידות זכרון בעלות תאים בני 8 סיביות. קשה לתאר את כל הצירופים הקיימים והאפשריים. יחידות הזכרון משתנות חדשות לבקרים ולכן נמנע מכך, אם כי יחידות הזכרון הנפוצות ביותר היום, הן בעלות תאים בני 8 סיביות.

קוי הבקרה קובעים את מצב החוצץ. לחוצץ שלושה מצבים אפשריים - יציאה (בקריאה מהרכיב), כניסה (בכתיבה ברכיב) ו-3-STATE כאשר לא פונים אל הרכיב. במצב זה החוצץ גורם לניתוק יחידת הזכרון ממסלול הנתונים. בהתאם לכך, מתחלקים קוי הבקרה לרכיב לשניים. קוים הקובעים READ (מהרכיב) או WRITE (ברכיב), וקוים הקובעים את הרכיב ב-3-STATE או במצב פעולה. לקוים הראשונים נתייחס בשני

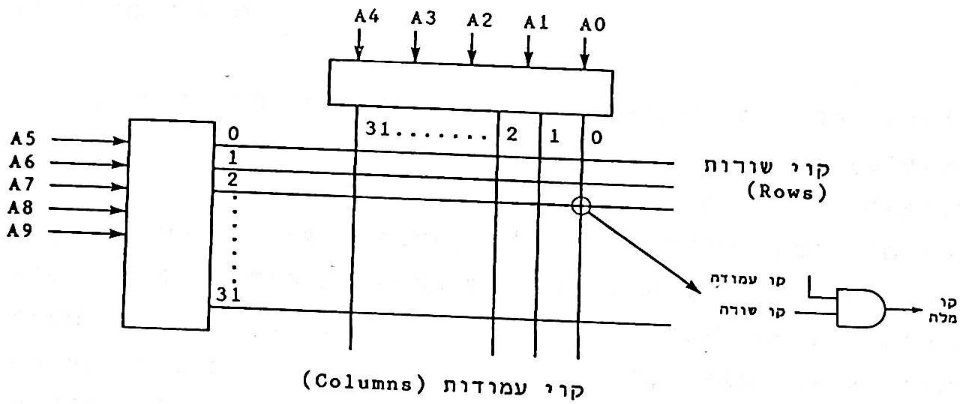
הסעיפים הבאים.

לקוים הקובעים את מצבו של הרכיב קוראים קוי 'CS (Chip Select - בחירת הרכיב). לעתים מסמנים אותם גם E (Enable - אפשר). מספר קוי ה-'CS שונה מרכיב לרכיב. גם לוגיקת הקוים משתנה. יש העובדים בלוגיקה חיובית ומסומנים CS, אבל רובם עובדים בלוגיקה שלילית ומסומנים 'CS. רכיב יוצא מ-3-STATE כאשר פעולת AND על קוי ה-'CS נותנת High. לדוגמא, הרכיב שבציור 2-13, RAM 6264 יוצא מ-3-STATE כאשר CS1=Low ו-CS2=High. כלומר, CS2.CS1'=High.



ציור 2-13 דוגמא ליחידת זכרון

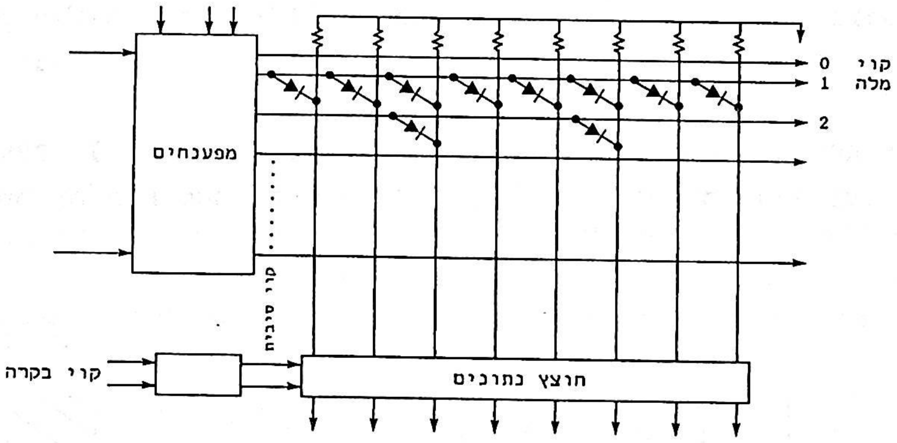
כתאר עתה את המבנה הפנימי של הזכרונות בעלי הגישה האקראית. גם ה-ROM לסוגיו הוא בעל גישה אקראית. המפענח המתואר בציור 2-14 מורכב למעשה משני מפענחים, על מנת להקטין את מספר היציאות. מפענח בעל 10 קוי כתובת בכניסותיו, צריך לכלול 1024 (2^{10}) קוי יציאה. דבר זה קשה למימוש מבחינה טכנית. פיצול המפענח לשני מפענחים בני 5 קוי כניסה ו-32 קוי יציאה בכל אחד קל יותר למימוש. התאים עצמם מאורגנים במערך בשורות ועמודות. קוי יציאה של מפענח אחד עוברים לאורך העמודות (Columns) וקוי היציאה של המפענח השני עוברים לאורך השורות (Rows) בצורה הבאה:



ציור 2-14 מבנה מערך תאי ROM

בכל צומת מפגש בין קו עמודה לקו שורה ישנו שער AND שכניסותיו הן קו עמודה וקו שורה. ליציאתו קוראים קו מלה. סך הכל יש 1024 צמתים ולכן 1024 קוי מלה. קוי המלה מגיעים אל תאי הזכרון. כל קו מלה מופעל רק כאשר מופיע הצירוף המתאים בעשרת קוי הכתובת. לכל קו מלה צירוף אחד ויחיד. כל קו מלה מפעיל תא זכרון אחד. תא הזכרון יכול להיות בן סיבית אחת או יותר. המקובל הוא בן 8 סיביות ואז קו המלה מגיע לכל שמונה הסיביות. מוצאי כל התאים זורמים אל חוצץ הנתונים דרך קוים הנקראים קוי סיבית. מבנה התא משתנה בהתאם לסוג הרכיב ROM , PROM, EPROM, RAM וכד'.

מערך תאי ה-ROM הוא מטריצת דיודות או מטריצת טרנזיסטורים המחוברים בצורה הבאה:



ציור 2-15 תיאור הדיודות במערך

כאשר קו מלה 1 מופעל, נקבל בכניסות החוצץ את המספר FFH (כל הדיודות נמצאות ולכן מאלצות "1"). כאשר קו מלה 2 מופעל, נקבל בכניסות החוצץ את המספר 24F (00100100) בהתאם לדיודות. במקום דיודות ניתן להשתמש בשערי MOSFET לאותה תכלית.

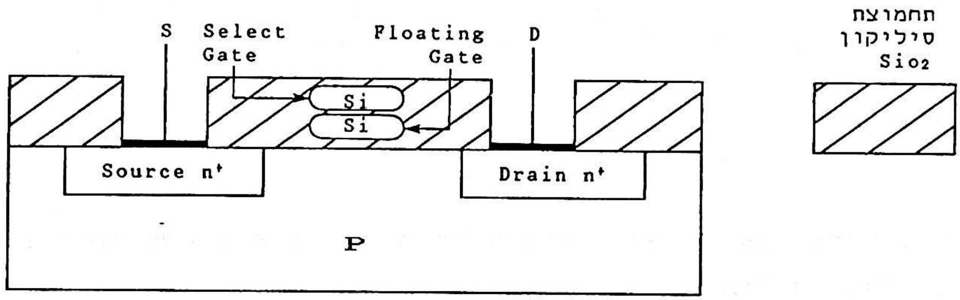
<u>מבחינים ב-4 סוגים עיקריים של ROM:</u>

MASK ROM - ROM מסיכה. זהו זכרון ROM שתוכן תאיו נקבע לפי הזמנת המשתמש במפעל שמייצר את הרכיבים. החיבורים של הדיודות במערך נעשים בעזרת מסיכת חיבורים בשלבי ייצור המעגל המשולב.

PROM (Programmable ROM) - ROM הניתן לתיכנות. ROM זה מגיע מבית-החרושת כשכל תאיו ריקים ממידע (בדרך כלל מלאים בתוכן FF). מערך הדיודות מלא, אבל כל דיודה מחוברת למערך דרך נתיך קטן. הכנסת המידע לתוכו נעשית על-ידי מסירת המידע לתאים בזה אחר זה, תוך כדי מסירת דופק מתח גבוה. דופק המתח יוצר דופק זרם גבוה דרך

הדיודה ששם המידע "0", הכתיך המחובר אליה ניתך ומידע זה נקבע. אין אפשרות לשנות את המידע ל-PROM שתוכנת (להעלות מצב סיבית חזרה ל-"1"). פעולת אחסנת המידע נקראת פעולת "צריבה".

EPROM (Erasable PROM) - PROM הניתן למחיקה. ב-ROM זה המטריצה היא מטריצת שערי MOSFET. הטרנזיסטור בנוי בצורה הבאה:



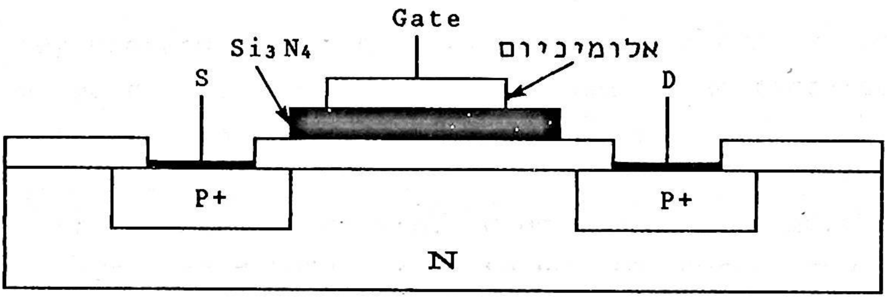
ציור 2-16 מבנה טרנזיסטור המיתוג ב-EPROM

גם כאן נעשית הצריבה בעזרת דופק מתח גבוה יחד עם מסירת מידע לתאים בזה אחר זה. הדופק הגבוה גורם, בטרנזיסטור שהמידע בו "0", למעבר מטענים מה-Select GATE אל השער הצף שבתחמוצת. שער צף זה נשאר טעון גם לאחר הפסקת הדופק וגורם לטרנזיסטור לא להוליך. הטרנזיסטור הוא מסוג אפיק N מועשר (בין ה-D ל-S קיימת תעלת N בתוך איזור ה-P). עם הימצאות אלקטרונים בשער הצף, נדחים האלקטרונים מאיזור האפיק; האפיק נעלם, והטרנזיסטור לא מוליך.

כדי למחוק את ה-EPROM (להחזירו למצב שבו כל תאיו FF והטרנזיסטורים מנותקים), מקרינים את ה-EPROM באור אולטרא-סגול. אור זה גורם לתחמוצת, שהיא מבדד מעולה בדרך כלל, להוליך והמטענים שבשער הצף חוזרים למקומם. הטרנזיסטורים חוזרים למצבם הראשוני, עד לצריבה הבאה. זו

הסיבה שברכיבי ה-EPROM מורכב חלון פלסטי שקוף, דרכו ניתן להקרין על הפיסה. כדי שתוכן ה-EPROM לא ימחק במקרה, אוטמים את החלון במדבקה, לאחר צריבתו. משך ההקרנה, תחת מנורת אולטרא-סגול חזקה, הוא כ-20 דקות. אי אפשר למחוק תא מסוים, אלא את כל תאי הרכיב.

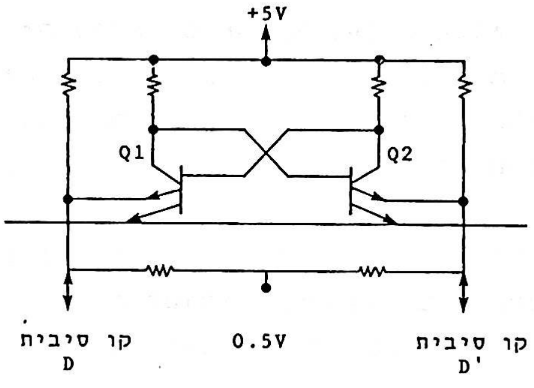
EEPROM (Electrical EPROM) - EPROM הניתן למחיקה בצורה חשמלית. ROM זה נקרא גם EAROM (Electrical Alterable ROM) - ROM הניתן לשינוי בצורה חשמלית. כשמו כן הוא. גם איחסון המידע וגם מחיקתו נעשים בצורה חשמלית. למעשה, ניתן להחליף לו את המידע, כאשר כותבים לתוכו תחת מתח גבוה. קיימים כבר EEPROMS הנצרבים תחת מתח של 5V . הוא אינו יכול לשמש כ-RAM, מכיוון, שמשך הכתיבה ארוך בהרבה ממשך הקריאה. טרנזיסטורי המיתוג ב-EEPROM, גם הם טרנזיסטורי MOSFET ובנויים בצורה הבאה:



ציור 2-17 מבנה טרנזיסטור המיתוג ב-EEPROM

בטרנזיסטור זה הוחלף השער הצף בשכבת סיליקון ניטריד (Si_3N_4) ומעליו שער אלומיניום. פולס הצריבה גורם לטעינת הסיליקון ניטריד ולהולכת הטרנזיסטור ופולס המחיקה גורם לפריקתו. טכנולוגיה זו נקראת MNOS. שים לב שהטרנזיסטור הוא מסוג אפיק P. ניתן להחליף תוכן של תא בודד.

תאי ה-RAM בנויים מתאי .D-F.F. קו המלה גורם לבחירת התא המבוקש. קו הסיבית מאפשר קריאת המידע הנמצא בסיבית התא המצוין. קו הסיבית מאפשר גם כתיבת מידע לתוך התא. מידע זה ישאר בתא עד לכתיבה הבאה. בזכרון ה-RAM יש מערכת כפולה של קוי סיבית. קו מידע וקו המשלים שלו - D ו-'D. צורת תא של סיבית אחת נראה בצורה הבאה:



ציור 18-2 תיאור תא RAM סטטי

לחוצץ הנתונים מגיעים קוי הסיביות D. כאשר נבחר קו מלה, הוא מקבל "1". אם חוצץ הנתונים במצב קריאה וטרנזיסטור Q1 ברוויה (Q2 בקטעון), יופיע "0" בקו הסיבית D. אם חוצץ הנתונים יהיה במצב כתיבה ויאלץ "1" בקו הסיבית D, זה יגרום למעבר Q2 לרוויה. לאחר שחרור הפליפ-פלופ הזה (קו מלה ירד ל-"0"), יאלץ Q2 את Q1 לעבור לקטעון. בקריאה הבאה מהתא הזה נמצא בקו הסיבית את המידע "1".

2.6 קביעת יחידת ROM בתחום כתובות

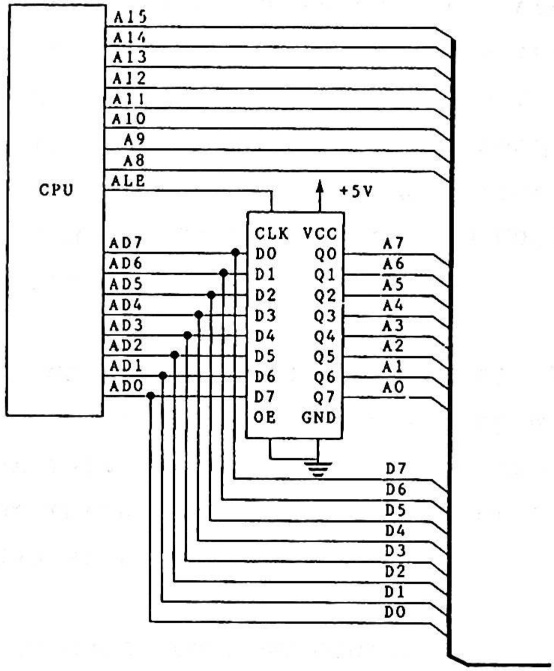
כמתואר בסעיפים קודמים, למיקרופרוסטורים של אינטל (ה-8085, ה-8051 וה-8088) יש מסלול מרובב (Multiplexed Bus). מסלול זה מכיל את הקוים AD0-AD7 ועוברים בו גם

כתובות וגם נתונים בזמנים שונים של מחזור המכונה. שיטה זו באה לנצל טוב יותר את רגלי ה-CPU. לאינטל יש רכיבים היודעים לזהות ולהפריד בין הנתונים לכתובות ולכן הם מתחברים ישירות ל-CPU. נכיר רכיב כזה בפרק הבא. בדרך כלל, לרכיבי התמיכה הסטנדרטיים של המיקרופרוססורים, כמו רכיבי זכרון (ROM ו-RAM) או פורטים, יש לספק קוי כתובת וקוי נתונים נפרדים.

בתחילת מחזור המכונה, מוציא ה-CPU בקוים AD0-AD7 את החלק הנמוך של הכתובת (A0-A7). הוא מסמן שזוהי כתובת בעזרת קו ALE (Address Latch Enable), העולה ל-High. זמן קצר לאחר ירידת קו ALE ל-Low, יזרמו נתונים בקוים AD0-AD7 מה-CPU או אל ה-CPU.

כדי לשמור על החלק הנמוך של הכתובת גם בחלק השני של מחזור המכונה, נועלים אותו ברכיב הנקרא Latch (נועל). רכיב כזה כולל שמונה תאי D-latch, בעלי כניסת clock משותפת שאליה מחברים את קו ALE. לכניסות התאים מחברים את הקוים AD0-AD7. כאשר קו ALE ב-High, יציאות התאים מקבלות את מצב כניסותיהם, כלומר, את הכתובת A0-A7. ירידת קו ALE ל-Low, נועלת בתאי ה-Latch את המצב בזמן הירידה והכתובת A0-A7 נשמרת עד לעליית קו ALE שוב ל-High, במחזור המכונה הבא. הזמן שבו קו ALE נמצא ב-High מאפשר לכתובת להתייצב בקוים, כך שבזמן הנעילה המידע יציב.

יציאות תאי ה-Latch משמשות כקוים A0-A7. תמיד יופיעו בהן רק אותות כתובת. הקוים AD0-AD7, היוצאים מה-CPU והמחוברים לכניסות התאים, משמשים לרכיבי התמיכה כקוי נתונים. למרות שמופיעים בהם גם אותות כתובת, אין זה מפריע כי רכיבי התמיכה מקבלים גם אותות המסמנים להם מתי להתייחס למסלול הנתונים. רכיב מקובל כ-Latch הוא הרכיב 74LS373. מעגל ההפרדה נראה בצורה הבאה:



ציור 2-19 הפרדת המסלול המרובב

מערכת מיקרו-מחשב מורכבת ממספר רב של קוים, המתחברים לרכיבים השונים. כדי להימנע משרטוט חשמלי, הכולל רשת צפופה של קוים לא ברורים נוהגים לאגד את הקוים בצמות כמתואר בציור. הדבר מחייב לציין שם לכל קו בהתאם לפונקציה שהוא ממלא.

ליחידת הזכרון המתחברת, נתייחס כאל יחידת זכרון אחת בעלת מספר תאים מוגדר וכל תא הוא בן 8 סיביות. יחידה כזו יכולה להיות מורכבת מכמה תת יחידות.

ליחידת זכרון כזו יש 8 קוי נתונים ו-n קוי כתובת בהתאם למספר התאים. מספר התאים שווה ל-2^n. לדוגמא, יחידת זכרון בת 10 קוי כתובת תכיל 1024 (2^{10}) תאים. ציון מספר בינארי מסוים בקוי הכתובת, מפנה לכיוון קוי הנתונים של יחידת הזכרון את התא שזהו מספרו.

כמצוין בפרק 1, מחוברים רכיבי התמיכה אל ה-CPU במקביל דרך מסלול הנתונים. בכל פניה של ה-CPU, מתחבר אל מסלול הנתונים רכיב תמיכה אחד בלבד. כדי לאפשר חיבור וניתוק כזה, יחידות הזכרון כוללות קו הנקרא 'CS (Chip Select). הסימן ' מציין עבודה בלוגיקה שלילית.

כאשר קו 'CS ב-Low, מחובר התא שכתובתו מצוינת בקוי הכתובת אל קוי הנתונים של הרכיב (ראה ציור 1-1). כאשר 'CS הוא ב-High, גם אם רשומה כתובת בקוי הכתובת, מנותקים התאים מקוי הנתונים. קוי הנתונים של הרכיב נמצאים במצב נתק הנקרא מצב של התנגדות גבוהה - High Impedance או מצב שלישי (3-STATE - לא "0" ולא "1"

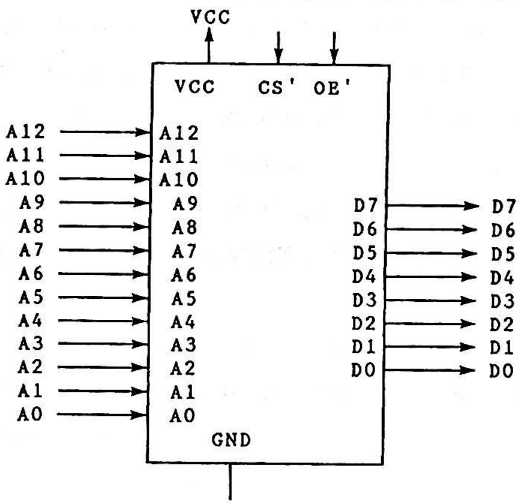
לרוב יחידות הזכרון יש קו נוסף לקו ה-'CS הנקרא 'OE (או 'OD). קו זה מיועד להקטנת צריכת הזרם של הרכיב, והקטנת ההשהייה של מעגל הפענוח.

כאשר מציין ה-CPU בקוי הכתובת כתובת של תא ביחידת הזכרון, לוקח זמן לכתובת זו להתייצב בקוים, עקב תופעות המעבר השונות. גם ליחידת הזכרון לוקח זמן לפענח את הכתובת ולהפנות את התא לכיוון יציאות הנתונים של היחידה. יחידת הזכרון כוללת במוצא חוצץ, המתמתג כאשר קו 'OE יורד ל-Low. קו 'CS יורד ל-Low כאשר ה-CPU פונה אל יחידת הזכרון. אין טעם להוריד ל-Low באותו זמן גם את קו 'OE, כי הכתובת אינה יציבה עדיין והנתונים שיגיעו אל חוצץ המוצא ישתנו. צריכת הזרם הגבוהה ביותר ברכיבים לוגיים נמדדת במצבי המעבר שלהם. מורידים את קו 'OE זמן

מה לאחר מתן הכתובת.

לקו 'OE מחברים את קו 'RD או 'PSEN (ב-8051), היורדים ל-Low כאשר ה-CPU מבקש לקרוא נתון מיחידת זכרון. אם נתבונן בדיאגרמות הזמנים נראה, שקוים אלה יורדים ל-Low זמן מה לאחר התחלת מחזור הקריאה וציון הכתובת, וזאת על מנת לאפשר התייצבות קוי הכתובת.

כתאר יחידת זכרון EPROM בת 8K (1K=1024) תאים בני 8 סיביות - 2764. יחידות ה-EPROM הסטנדרטיות מתחילות בספרות 27 והספרות שבהמשך מציינות את מספר הסיביות (64K bit). יחידה כזו תיראה בצורה הבאה:



ציור 2-20 יחידת EPROM 8K*8 bit

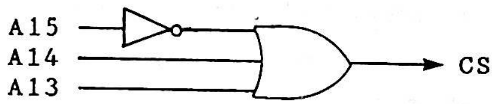
כדי לקבוע את יחידת הזכרון בתחום כתובות מסוים, מפענחים בעזרת מעגל פענוח את סיביות ה-MSB שאינן מחוברות אליה.

נמקם את יחידת הזכרון שמופיעה בציור הקודם בתחום הכתובות 8000-9FFF. בתחום זה יש 8192 כתובות. בדוק זאת. כל פניה לכתובת בתחום כתובות זה תיצור צירוף כלשהו

בקוים A0-A12 אבל הקוים A13 עד A15 יציגו את המספר הבינארי 100.

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
8000H =	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
כתובת בתחום =	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X
9FFFH =	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

מעגל הפענוח יפענח את הקוים A15-A13. יציאתו תרד ל-Low כאשר בכניסותיו יופיע הצירוף 100. מעגל כזה יראה בצורה הבאה:

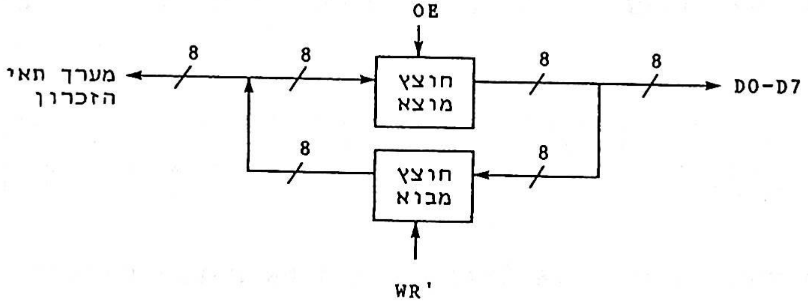


ציור 2-21 מעגל פענוח לזיהוי 100

יחידת הזכרון המחוברת לקו 'CS תמותג רק כאשר יתקבל בקוים A15-A13 הצירוף המבוקש (100).

2.7 <u>קביעת יחידת RAM בתחום כתובות</u>

יחידת RAM בנויה באותה המתכונת כמו יחידת ה-ROM. גם היא כוללת n קוי כתובת עבור 2^n תאים, קו 'CS וקו 'OE, המחובר לחוצץ המוצא. בנוסף לחוצץ המוצא, כולל ה-RAM גם חוצץ מבוא, שכניסותיו מחוברות לקוי הנתונים, ויציאותיו לכניסות חוצץ המבוא. חוצץ זה מופעל בעזרת קו מיוחד, הנקרא 'WR. ניתן לראות חוצץ זה כחוצץ דו כיווני.

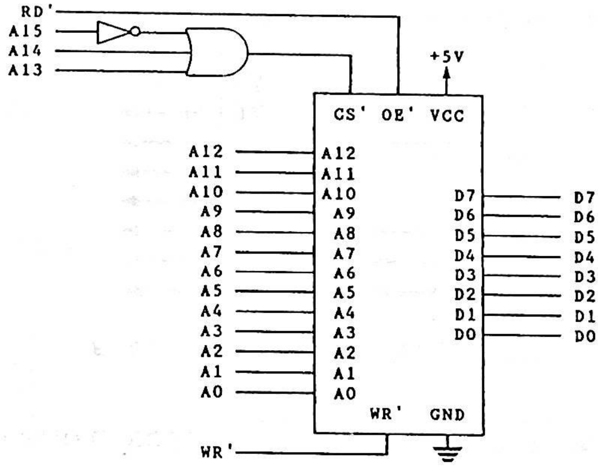


ציור 2-22 תיאור חוצצי ה-RAM

כאשר קו 'OE יורד ל-Low וקו 'WR נמצא ב-High, מופעל חוצץ המוצא בלבד, והנתונים זורמים ממערך תאי הזכרון אל קוי הנתונים D0-D7. כאשר קו 'WR יורד ל-Low וקו 'OE נמצא ב-High, מופעל חוצץ המבוא בלבד, והנתונים זורמים מקוי הנתונים D0-D7 אל מערך תאי הזכרון.

לקו 'WR מחברים את קו 'WR של ה-CPU. כאמור בסעיף הקודם, מחברים לקו 'OE את קו 'RD או 'PSEN (ב-8051). ה-CPU דואג שלעולם לא יקרה מצב שבו גם קו 'RD (או 'PSEN) וגם קו 'WR יהיו ב-Low באותו זמן (יתכן ששניהם יהיו ב-High), ולכן שני החוצצים לעולם לא ימותגו בו זמנית.

קביעת יחידת RAM בתחום כתובות מסוים נעשית בדומה לקביעת יחידת ROM. כל ההבדל הוא בחיבור קו 'WR של ה-CPU לרגל 'WR של הרכיב. לדוגמא, חיבור יחידת 8K*8 RAM (6264) לתחום הכתובות 8000-9FFF. מעגל הפענוח נלקח מהדוגמא מהסעיף הקודם של חיבור ה-ROM.



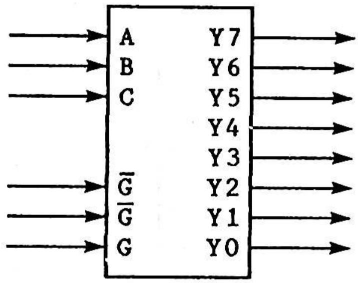
ציור 2-23 קביעת RAM 6264 בתחום כתובות

2.8 בניית מערך יחידות זכרון

את מעגל הפענוח ניתן לממש בצורות שונות. אחת השיטות לכך היא בעזרת מפענחים. בשיטה זו מקבלים למעשה מספר קוי 'CS, כאשר כל אחד הוא עבור תחום כתובות אחר. חיבור יחידות זכרון לקוי 'CS אלה יוצר מערך תאי זכרון על תחום כתובות רחב.

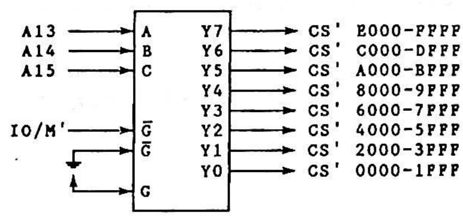
רכיב פופולרי מאד כמפענח הוא הרכיב 74138. זהו מפענח 1 מתוך 8, הכולל 3 כניסות כתובת A,B,C ו-8 יציאות. כל צירוף בינארי בכניסות הכתובת גורם להופעת "0" באחת היציאות. אין מצב שבו שתי יציאות ב-"0". בנוסף לכניסות הכתובת, כולל הרכיב 3 כניסות איפשור (Enable), שתיים בלוגיקה הפוכה.

74138

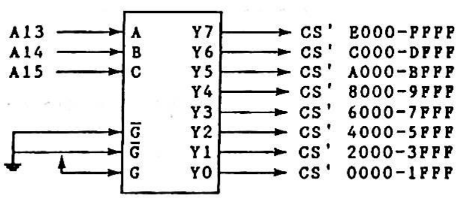


ציור 2-24 תיאור 74138

רק כאשר מופיעים בכניסות האיפשור (G,G,G) הצירוף 0,0,1 בהתאם, יורדת אחת הכניסות ל-"0" בהתאם לצירוף הבינארי בכניסות C,B,A (A הוא ה-LSB). אם נחבר לקוים C,B,A את קוי הכתובת A13, A14, A15, נקבל שמונה קוי CS, שכל אחד ירד ל-"0" בתחום כתובות אחר. ב-8085 וב-8088 נחבר לאחד מקוי ה-G, את הקו IO/M' על מנת לאפשר CS רק במקרה של פניה לזכרון.



ציור 2-25 פענוח עבור 8085 ו-8088

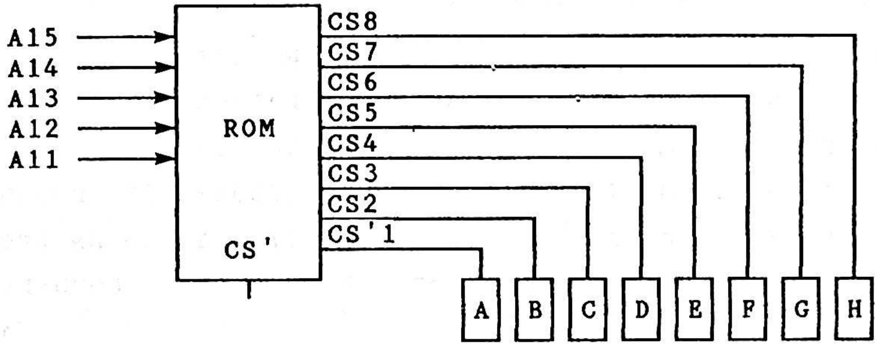


ציור 2-26 פענוח עבור ה-8051

לא חייבים להשתמש במפענח רק כאשר זקוקים לכמה קוי בחירת רכיב (CS). נוח וכדאי להשתמש בו גם במקרה של יחידת זכרון אחת. אם נתבונן במעגל הפענוח מהסעיף הקודם, נראה שאנו זקוקים לשני רכיבים למימושו. ה-74138 מאפשר מימוש מעגל פענוח בעזרת רכיב אחד בלבד. ניתן גם לשנות את תחום הבחירה בקלות יחסית.

שיטה אחרת לפענוח מלא היא פענוח בעזרת ROM. בשיטה זו, מחברים את יציאות הנתונים של ה-ROM לכניסות ה-'CS של הרכיבים השונים. בתאי ה-ROM קובעים נתונים המוציאים 'CS בהתאם לקוי הכתובת של הרכיבים.

לדוגמא, מיתוג 8 רכיבי זכרון בעזרת ROM בן 32 תאים (5 כניסות כתובת) בני 8 סיביות כל אחד.

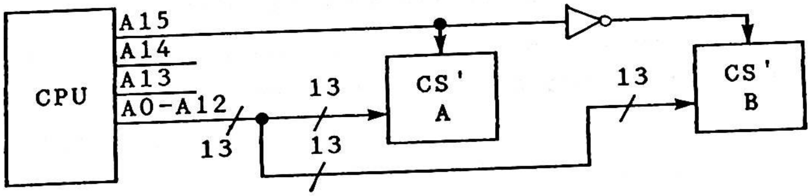


ציור 2-27 ROM כמפענח יחידות זכרון במערך

בשיטה זו של פענוח, ניתן לקבוע בעזרת רכיב אחד את כל מפת הזכרון, גם כאשר יחידות הזכרון ממותגות לא בתחום כתובות רציף.

במערכות מיקרו-מחשב קטנות שאינן דורשות יחידות זכרון גדולות, וברור גם שלא תידרש הרחבת הזכרון, ניתן לצמצם את מעגל הפענוח או אפילו לוותר עליו.

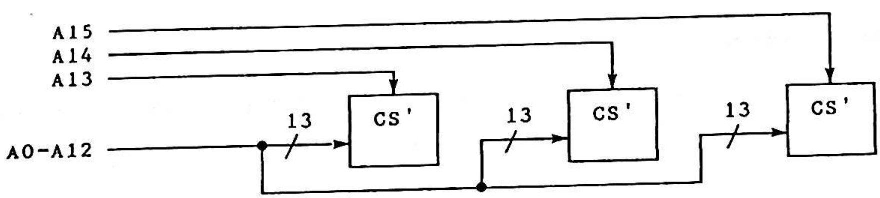
אם למשל, נחבר ל-CPU שתי יחידות זכרון בנות 8K כל אחת, נוכל לחברן כמתואר בציור הבא.



ציור 2-28 פענות חלקי של קוי הכתובת

בצורת חיבור זו, אין מפענחים את הקוים A11-A14. כאשר A15 ב-Low, ממותגת היחידה A ולכן היא תופסת את תחום הכתובות 0000-1FFF. אבל, באותה מידה היא תופסת את הכתובות 2000-3FFF, 4000-5FFF ו-6000-7FFF. יחידה B תופסת את הכתובות 8000-9FFF וכן את התחומים A000-BFFF, C000-DFFF ו-E000-FFFF. כותב התוכנה צריך להיות מודע לכך.

שיטה אחרת לפענוח מינימלי ללא שערים כלל, הנקראת גם פענוח ליניארי, מתוארת בציור 2-29.



ציור 2-29 פענוח ליניארי של קוי הכתובת

בשיטה זו, שאינה דורשת שערים או מפענחים יש להיזהר מאד. הסיבה לכך היא שישנם תחומי כתובות בהם ממותגים שני

רכיבים למסלול הנתונים, ויש להיזהר מפניה לתחומי כתובות אלה.

בפענוח כתובות יש להתייחס להיבט נוסף. בזמן פעולת ה-CPU, כמתואר בפרקים קודמים, ישנם מצבים בהם ה-BUS פנוי. במצבים אלה, קוי כתובת במצב לא מוגדר, כלומר, הכתובת הרשומה בהם היא אקראית ומשתנה. מבחינת ה-CPU אין בכך כל רע. בלאו הכי, הוא אינו מצפה לנתון כלשהו במסלול הנתונים. הבעיות הינן מבחינת רכיבי התמיכה.

א. אם שיטת הפענוח היא ליניארית, יתכן מצב שהכתובת האקראית הינה כתובת המתאימה לשני רכיבי תמיכה ואז שניהם יפנו מידע שונה למסלול הנתונים ויעמיסו זה את זה.

ב. צריכת הזרם העיקרית ברכיב היא בזמן המעבר של היציאות מ-High ל-Low ולהיפך. כאשר משתנות הכתובות אקראית, מחליפות היציאות מצב וצריכת הזרם במערכת עולה.

ג. ב-8085, במצב של BUS פנוי, יתכן שיחידת זכרון ויחידת I/O ימותגו בו זמנית ויפנו מידע שונה למסלול הנתונים.

זו הסיבה שרכיבי התמיכה או מעגלי הפענוח נעזרים בפענוח בקוי הבקרה 'RD, 'WR ו-'PSEN. קוים אלה יורדים ל-Low רק בזמן שה-CPU פונה באמת לאחת היחידות ובכל מקרה רק לאחת מהן באותו זמן.

2.9 זיהוי דיאגרמת זמנים

תנאי הכרחי לאיתור תקלות הוא זיהוי והכרת דיאגרמות הזמנים של ה-CPU. איתור תקלות בחומרה במערכת מיקרו-מחשב קשה, מכיון שהמיקרו-מחשב הוא מערכת דינאמית. קשה

להסתנכרן על פולסי השעון שלו ולזהות את הבעיה בעזרת משקף תנודות (אוסצילוסקופ) אבל אפשרי. קיימים אמצעים מתוחכמים ויקרים לאיתור תקלות, אבל לא נעסוק בהם. בעיה נפוצה היא, שבפניה לכתובת מסוימת, אין ה-CPU מקבל את הנתון הדרוש, או הנתון שהוא מוסר איננו מגיע ליעדו מסיבה כלשהי. אם הנתון המבוקש היה אמור להיות הוראה מסוימת (חלק מתוכנית המחשב), והגיע שגוי, כל התוכנית משתבשת. במקרה כזה, אי אפשר לעקוב ולחזות את מהלכי ה-CPU.

זיהוי דיאגרמת זמנים באמצעות סקופ, מחייב להביא את ה-CPU לפעולה מחזורית. מריצים אותו על תוכנית הפונה בלולאה אינסופית לכתובת הבעייתית או המיועדת להיבדק. התוכנית צריכה להיות קצרה על מנת לראות את כל המחזור בסקופ. בצורה כזו ניתן להסתנכרן על אחד האותות, ולעקוב אחר השתנות האותות האחרים בהתאמה.

לדוגמא, נניח שהכתובת הנבדקת נמצאת בכתובת 8050H. תוכנית לפניה לכתובת זו בלולאה אינסופית תיראה בצורה הבאה:

8085

```
LOOP: LDA 8050
      JMP LOOP
```

8051

```
      MOV  DPTR,#8050
LOOP: MOVX A,@DPTR
      SJMP LOOP
```

8088

```
LOOP: MOV AL,[8050]
      JMP LOOP
```

הלולאה מתבצעת בעזרת שתי פקודות בלבד, אבל התוכנית מורכבת ממספר בתים. ה-CPU מבצע Fetch של בתי הוראה ולאחר מכן מבצע. כלומר הלולאה תכלול מספר מחזורי מכונה של קריאת בתים על-ידי ה-CPU, אבל רק מחזור מכונה אחד יכיל פניה לכתובת 8050. כל שאר המחזורים יכילו קריאת בתים מכתובות באיזור התוכנית.

2.10 איתור תקלות בחומרה

כפי שנאמר בסעיף הקודם, השיטה באיתור תקלה בחומרה, היא הרצת ה-CPU בלולאה אינסופית כאשר בתוך הלולאה הוא פונה פעם אחת לכתובת הבעייתית. יציאת מעגל הפענוח מספקת אות סינכרון המסמן מתי ה-CPU פונה לכתובת זו.

קו 'CS של מעגל הפענוח ימצא ב-Low רק כאשר תופיע בקוי הכתובת כתובת מתאימה. בכל כתובת אחרת ימצא קו 'CS ב-High. כדי לבדוק את זרימת האותות על קוי המסלולים בפניה לכתובת מסוימת, יש לבצע את השלבים הבאים:

א. הרצת תוכנית הפונה במהלכה פעם אחת בלבד לכתובת זו. הדבר ייצור אות מחזורי המופיע פעם אחת בכל לולאה של התוכנית.

ב. חיבור אחד מבחוני המשקף ליציאת מעגל הפענוח והסתנכרנות על אות זה.

ג. חיבור הבחון השני של המשקף לקוי המסלולים בזה אחר זה ובדיקת מצבם בהתאמה לאות המוצא של קו הפענות. ניתן להשוות בין האותות המופיעים על רגלי ה-CPU לאותות המופיעים על הרגלים המתאימות של יחידות הזכרון.

לראות דיאגרמת זמנים בעזרת סקופ בעל שני ערוצים, זה עסק די מייגע. גם כאשר מסונכרנים לקו ה-'CS, קשה לזהות בדייקנות את מחזורי השעון השונים ולייחס אליהם את מצבי האותות. למרות זאת זה ניתן.

מחברים את הבחון השני לקו CLK (הראשון מחובר קבוע לקו 'CS). מכיילים את הסקופ (בעזרת כפתור השינוי הרציף במחולל בסיס הזמן) כך שכל מחזור שעון יותאם לחלוקה מדוייקת על ציר X (2 מ"מ או 3 מ"מ). בצורה כזו נקבל

כיול של ציר X בהתאם למחזורי השעון. לאחר מכן נוכל לצייר את האותות השונים בהתאם למחזורי השעון.

לצורך זיהוי דיאגרמת זמנים מקובל להשתמש במכשיר הנקרא נתח לוגי. זה מכשיר הכולל מספר רב של בחונים, אותם מחברים אל נקודות הבדיקה השונות. כאשר מפעילים אותו, הוא מציג במסך את דיאגרמת הזמנים של כל האותות הנדגמים, בבת אחת. איננו מסוגלים לראות תופעות מעבר בעזרתו כי הוא מציג את האותות בצורה לוגית ("0" או "1"). הנתח הלוגי הוא כלי חשוב, אבל לא נעסוק בו בספר, מכיוון שאין באפשרותנו להתייחס לנתח לוגי מסוים.

אפשרות אחרת להקל על זיהוי דיאגרמת זמנים הוא שימוש בסקופ רב ערוצים. לסקופ זה יש יותר מ-2 ערוצים (שמונה או יותר). בעזרת ערוצים אלה ניתן לראות מספר רב של ערוצים במקביל.

ייתכן מאד, שבמחזור לולאה אחת, תופיע בתצוגת הסקופ, ירידה של קו ה-'CS ל-Low גם במקום נוסף, שמבחינת התוכנית אין בו פניה אל הרכיב. הדבר לא נובע בהכרח מפגם כלשהו. ישנם מחזורי מכונה או מחזורי שעון מסוימים, בהם מבצע ה-CPU פעולות פנימיות כלשהן. במקרה כזה הכתובת בקוי הכתובת היא כתובת אקראית. לעתים קרובות זוהי כתובת שהוצאה קודם, ולכן יופיע 'CS נוסף. הדבר איננו מפריע לרכיבי התמיכה, כי במצב זה אין הקוים 'RD, 'PSEN או 'WR יורדים ל-Low, ולכן הם לא יתייחסו לירידה זו. זו סיבה טובה לכייל את התחום בין קוי ה-'CS בעזרת קו 'RD (או 'PSEN ב-8501) ולא בעזרת ה-CLK.

קו זה יציין לנו שני דברים חשובים. הוא יסמן את מחזורי המכונה השונים (מחזורי השעון פחות חשובים לנו, בייחוד כאשר מחזורי המכונה הם בעלי אורך משתנה). הוא יסמן את הזמן המדויק בו יש להתייחס אל הכתובת ואל הנתונים.

בירידת הקו ל-Low, הכתובת היא אמיתית ויציבה. בעליית הקו מ-Low ל-High או מעט לפני העליה, הנתון אמיתי ויציב. כאשר מתבוננים באותות הרועשים הנראים בסקופ, מבינים את חשיבות הפיסקה הזו. קו זה לא ירד ל-Low במחזור כתיבה, או מחזור של פעולה פנימית. קו 'PSEN לא ירד ל-Low גם במקרה של קריאה מ-RAM חיצוני.

דבר מעניין נוסף, שאנו עשויים להבחין בו, במיקרופרוססורים 8051 ו-8088 הוא תופעת ה-Pipe Line (ציגור). אנו עשויים לראות פניה לכתובת וקריאת נתון הנמצא מעבר לכתובות התוכנית. הסיבה לכך היא, שמיקרופרוססורים אלה מבצעים הבאת ההוראה הבאה בתוך במהלך ביצוע ההוראה הנוכחית. ה-8088 מבצע זאת בשלמות וה-8051 מבצע זאת חלקית. בכל מחזור מכונה הוא פשוט קורא 2 בתים. רק לאחר הבאת ההוראה הוא מפענח ומבצע. אם הוא מזהה פקודת הסתעפות, הוא מתעלם מהנתון הנוסף שהביא, אבל מחזור קריאה זו מופיע בדיאגרמת הזמנים.

<u>תקלות אופייניות במערכת מיקרו-מחשב:</u>

א. נתקים - קו היוצא מרכיב אחד ואיננו מגיע אל רגל אחד הרכיבים שאליו הוא מיועד להגיע. תקלה זו יכולה לנבוע מנתק במוליך או אפילו מהלחמה קרה. בדיקות ומדידות יש לעשות על רגלי הרכיבים עצמם ולא על נקודות ההלחמה, מכיון שיכול להיות שאל הנקודה זה מגיע, אבל ממנה אל הרגל עצמה יש נתק. בדרך כלל, ברגל כניסה הנמצאת באויר נמצא מתח DC באמצע התחום 0-5V, כלומר בסביבות 2.5V.

ב. קצר לאדמה - מוליך או רגל המקוצרים לאדמה. במקרה כזה מאולץ בנקודה 0V קבוע. הדבר מאלץ 0V גם ביציאה של הרכיב המספק את האות. דבר העלול לגרום לו נזק, אם כי לא תמיד. רוב הרכיבים מוגנים, כך שניתוק הקצר מחזיר אותם לכשירות.

ג. קצר ל-VCC - מאלץ את הרגל או את הקו להיות ב-0V. אם במערכת יש מקורות מתח מעל 5V לדוגמא +12V או -12V, אזי קצר למתח כזה יגרום בדרך כלל להינזקות הרכיבים.

ד. קצר בין קוים - כאשר שני קוים מקוצרים ביניהם, נקבל מתח באמצע התחום 0-5V כלומר 2.5V בערך. אם כי בדרך כלל ה-"0" חזק יותר. כאשר בקו אחד צריך להיות "0" ובקו השני צריך להיות "1", מה שנראה, בדרך כלל, שבקו המקוצר יהיה "0". קצר כזה יכול להיות בין קוי כתובת, בין קוי נתונים ובעצם בין כל זוג קוים אפשריים.

קצר בין קוי נתונים ניתן לזיהוי ואיתור בצורה לוגית. אנו מבחינים בו, קודם כל, כאשר כותבים ל-RAM נתון מסוים וקוראים מה-RAM נתון אחר. לדוגמא, אם קיים קצר בין קו D3 ל-D5. כל מספר בינארי שנכתוב שבו D3 או D5 שווים ל-"0", יגרום ששתי הסיביות תהיינה ב-"0",. רק אם שתיהן תהיינה ב-"1" נקבל "1" בשתיהן. כדי לאתר תקלה מסוג זה, כותבים ב-RAM את המספרים הבאים:

01 - 00000001
02 - 00000010
04 - 00000100
08 - 00001000
10 - 00010000
20 - 00100000
40 - 01000000
80 - 10000000

אם לדוגמא, D3 מקוצר ל-D5 נקרא מה-RAM (לאחר כתיבת המספרים המצוינים לעיל) את המספרים הבאים:

01 ,02 ,04 ,00 ,10 ,00 ,40 ,80

מכאן נוכל להסיק ש-D3 מקוצר ל-D5.

כאשר כותבים ל-RAM את המספר FF וקוראים "0" באחת הסיביות, ניתן להניח שאותה סיבית מקוצרת ל-GND. לא בהכרח ישירות ל-GND, אלא ייתכן שהיא מקוצרת לקו המאלץ בה "0".

כאשר כותבים ל-RAM את המספר 00 וקוראים "1" באחת הסיביות, ניתן להניח שאותה רגל מקוצרת ל-VCC או לקו המאלץ בה "1", או שהיא מנותקת. רגל באויר מתנהגת, בדרך כלל, כ-"1". אם כי, לא תמיד. ברכיבי MOSFET ההתנהגות היא התנהגות משתנית.

קצר בקוי כתובת או נתק של קוי כתובת גורם למיפוי כפול. במיפוי כפול, הכוונה שלכתובת מסוימת מגיעים בפניה לשתי כתובות או יותר. למשל, נניח שקו A3 מקוצר לאדמה ואנו פונים לשתי הכתובות 4000H ו-4008H:

4000H = 0100,0000,0000,0000

4008H = 0100,0000,0000,1000

בגלל ש-A3 מקוצר לאדמה, נקבל בקוי הכתובת בפניה ל-4008H את הכתובת 4000H. כלומר, נפנה בפועל שוב לכתובת 4000H. אם הקו היה מנותק או מקוצר ל-VCC, היינו מקבלים בשני המקרים פניה לכתובת 4008H. קיצור בין שני קוי כתובת גורם לאיפוס הקוים כאשר הקוים שונים זה מזה (אחד "0" והאחר "1"). כלומר, מגיעים לכתובת שבה שניהם ב-"0". רק כאשר שניהם שווים, מגיעים לכתובת נכונה. צריך לזכור שתופעה זו חוזרת בתחומי כתובות שונים. נחזור לדוגמא שבה A3 מקוצר לאדמה. נקבל את ההתנהגות הבאה:

4008H -> 4000H   4018H -> 4010H

4009H -> 4001H   4019H -> 4011H

400AH -> 4002H   401AH -> 4012H

וכך הלאה. כלומר פניה לכתובת בתחום 4008-400F תפנה אותנו לכתובת מקבילה בתחום 4000-4007. אותו הדבר לגבי כתובת בתחום 4018-401F, שתפנה אותנו לכתובת בתחום 4010-401F. המיפוי הכפול במקרה זה ממשיך בתחומים של 8 כתובות בכל פעם.

הצרה במיפוי כפול שאיננו יכולים לזהות אותו בפניה לכתובת מסוימת. בפניה לתא מסוים, איננו יודעים להיכן הגיע הנתון בפועל. אם נכתוב נתון בתא 4008H, ונקרא את הנתון מכתובת זו נבחין שאכן הנתון נמצא ב-RAM. זאת למרות שהוא נמצא בפועל בתא 4000H.

השיטה בבדיקת מיפוי כפול, היא מילוי תחום הכתובות הנבדק במספרים גדלים. לדוגמא, נניח שברשותנו RAM בתחום הכתובות 8000H-9FFFH. אנו רוצים לבדוק האם קיים בו מיפוי כפול. רושמים תוכנית, הממלאה אותו במספרים 00-FE ברציפות. לאחר מילוי הבלוק, ניתן להתבונן בו בעזרת Debugger, או להריץ תוכנית הקוראת תא אחר תא ובודקת האם זה הנתון שמצאנו. אם נחזור לדוגמא שבה A3 מקוצר לאדמה, נמצא בכתובת 8000H את המספר 08, שנכנס לשם כאשר פנינו לכתובת 8008. זאת למרות העובדה, שכשפנינו לתא 8000 נכנס לשם הנתון הנכון.

הסיבה שכתבנו מספר 00-FE ולא עד FF, היא כדי לא ליצור בלוקים שווים בתחומים שווים. לדוגמא, אם במקרה, היה A8 מקוצר לאדמה או ל-VCC, והיינו ממלאים את ה-RAM בנתונים 00-FF. בכתובת 8100H היינו רושמים 00. זה היה נכתב בפועל בתא 8000H. זה גם הנתון שימצא בתא 8000H ולכן לא נזהה תקלה זו. כתיבת המספרים 00-FE גורמת לכך שלכל גוש תאים שעלולים להיות במיפוי כפול, לא יוכנסו מספרים זהים.

כדי לבדוק מיפוי כפול ב-ROM, לוקחים ROM, המלא כבר במספרים האמורים, קוראים אותו ומשווים את המספרים הנקראים עם הנתונים הדרושים.

ייתכן מצב שקווי הכתובת הבעייתים הם מאלה המתחברים למעגל הפענוח ולא לרכיב עצמו. במצב כזה ייתכנו שני מקרים. מקרה אחד שבו הרכיב לא מקבל CS בפניה אליו. אז יש לבדוק בעזרת סקופ (לאחר הרצת תוכנית הפונה לרכיב בלולאה אינסופית) את האותות המגיעים למעגל הפענוח.

מקרה שני הוא שהרכיב מקבל CS. במבט ראשון המערכת נראית תקינה. אבל ייתכן שהוא מקבל CS גם במקרה של פניה לתחום כתובות מחוץ לתחום הכתובות של הרכיב. אם קיים רכיב בתחום הנוסף, נקבל מיפוי כפול בין רכיבים. שני הרכיבים משבשים זה את זה.

אין נוסחה ושיטה אחידה לאיתור תקלות בחומרה. מספר התקלות האפשריות הוא גדול. המאתר חייב להתבונן היטב בתוצאות המתקבלות ולהפעיל את הגיונו על מנת לאתר תקלה.

2.11 שאלות ותרגילים

1. שרטט דיאגרמת זמנים מפורטת, הכוללת את אותות השעון של ה-CPU (בו אתה משתמש), קוי הבקרה, 8 קוי הנתונים וקוי הכתובת, שמתארת את מצב הקוים כאשר ה-CPU כותב את הנתון A5 בכתובת C35B.
2. שרטט דיאגרמת זמנים מפורטת כמו בשאלה 1, שמתארת את מצב הקוים כאשר ה-CPU קורא נתון מהכתובת DB96. בכתובת זו נמצא הנתון 6E.
3. תכנן חיבור EPROM 27128 (16K*8) לתחום הכתובות 8000-BFFF. ממש את מעגל הפענוח בעזרת שערים. שרטט גם את ה-Latch להפרדה בין קוי הכתובת לקוי הנתונים.
4. תכנן חיבור RAM 6264 (8K*8) לתחום הכתובת C000-DFFF. ממש את מעגל הפענוח בעזרת המפענח 74138. מה עליך לעשות כדי לשנות את מיפוי ה-RAM לתחום הכתובות A000-BFFF?
5. רשום תוכנית המאחסנת בתא C35B (ב-RAM שתכננת בשאלה 4) את הנתון A5 בלולאה אינסופית. התוכנית מתחילה בכתובת 4000H (אם הנך משתמש ב-8051 או ב-8088) או בכתובת 2000H (אם הנך משתמש ב-8085).
6. מלא את הטבלה הבאה בצורה בינארית עבור כל אחד ממחזורי המכונה של הלולאה.

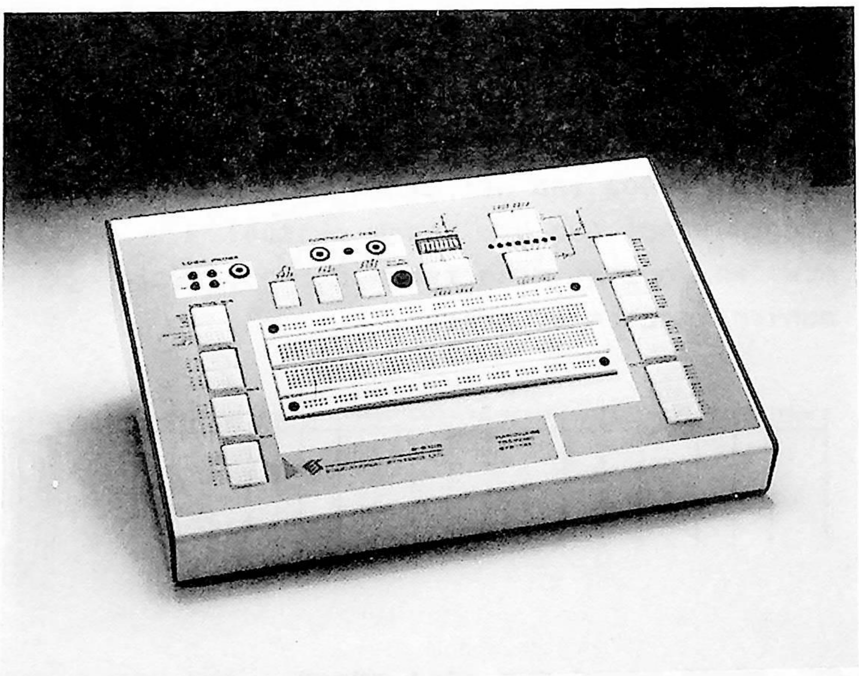
מחזור מכונה	A15	A14	A13	A12	A11	A10	A9	A8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	ALE	PSEN'	RD'	WR'	IO/M'
1a																					
1b																					
2a																					
2b																					
3a																					
3b																					

כל מחזור מכונה תאר בשני שלבים. פעם אחת (מצב a) כאשר קו ALE=1 ופעם שניה (מצב b) כאשר אחד מקוי הבקרה 'RD, 'WR או 'PSEN ב-"0".

7. הנך כותב נתון ב-RAM, שחיברת בתחום הכתובות C000-DFFF, וקורא אותו מה-RAM. מתברר לך שהנתון שקראת איננו הנתון שכתבת. הסבר מה יהיו פעולותיך לאיתור התקלות. אפשר לתאר זאת בעזרת תרשים זרימה.

2.12 תיאור הערכה S-2100 ומעגל DSM-2200

לצורך הניסויים בחומרה וברכיבים היקפיים, נשתמש בערכה S-2100 ובכרטיס DSM-2200. ערכה זו ניתנת לחיבור ישירות (בעזרת כבל רב גידי שטוח) לערכות S-2051, S-2085 או ל-APPLIC-1 לשם תרגול עם המיקרופרוססורים 8051, 8085 או 8088 בהתאמה. לכל אחת מהערכות יש יציאת BUS המוגנת על-ידי חוצצים והמספקת קוי כתובת, נתונים, בקרה ואספקת מתח. BUS זה נקרא SIM-BUS. מ-S-2051 ומ-S-2085 יוצא BUS מלא, הכולל ששה-עשר קוי כתובת (A0-A15) ומ-APPLIC-1 יוצא I/O-BUS בלבד הכולל 8 קוי כתובת (A0-A7).



הערכה S-2100 כוללת לוח תקיעה ראשי לעבודה בחיווט ללא הלחמות. משמאל הערכה נמצאים לוחות תקיעה קטנים המספקים את אותות ה-BUS המגיעים מהערכה המפעילה. מימין הערכה נמצאים ארבעה פורטים בנויים (עליהם נדבר בהמשך) לשם תרגול יישומים של יחידות קלט/פלט. בחלק העליון של הערכה נמצאים אביזרי עזר שונים. בחון לוגי (המשמש לזיהוי High, Low, 3-State וגל ריבועי), בודק רציפות, 4 יציאות מתחים (+5V, +12V, -12V, GND), 2 מתנדים, האחד עם תדר קבוע 50Hz והשני עם תדר הניתן לשינוי, 8 מפסקים זעירים ו-8 לדים. לכל כניסה או יציאה בערכה 4 נקודות חיבור. הערכה כולל ספק כח משלה ומפסק הפעלה ראשי.

חלק מהניסויים מתבצעים על לוח התקיעה הראשי. אבל ניסויים בחומרה כוללים חיווט רב. כדי להקל על ביצוע הניסויים מצורף לערכה הכרטיס DSM-2200. מעגל זה כולל את מרבית הרכיבים, כאשר הם מחוברים כבר למעגלי הפענוח. הם מסודרים בהתאם לניסויים השונים ומוסברים בעזרת הדפס לבן. מרבית הניסויים בספר זה יתבצעו בעזרת הכרטיס DSM-2200. הכרטיס כולל, בחלק התחתון שלו, פיני חיבור לערכה. ארבעה פינים להתחברות לנקודות אספקת המתחים ו-32 פינים להתחברות לקוי ה-BUS.

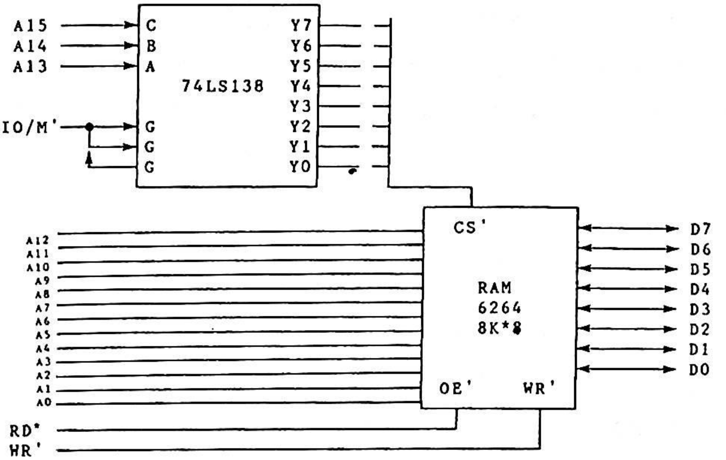
ניתן להפעיל את מעגל DSM-2200 ללא S-2100 בחיבור ישיר לערכות. בצורה זו יחסרו אביזרי העזר של הערכה S-2100 והמתרגל ידרש לאלתר מעט. ההתייחסות בספר היא לעבודה משולבת של DSM-2200 עם S-2100.

חבר בזהירות את הכרטיס DSM-2200 לערכה S-2100. חבר את S-2100 לערכה המפעילה (S-2051, S-2085 או APPLIC-1). הפעל את שתי הערכות.

2.13 חיבור RAM בתחום כתובות

ניסוי זה לא יכול להתבצע בעזרת IBM PC ו-APPLIC-1.

במעגל DSM-2200 תמצא את הרכיב 6264 המהווה RAM 8K*8.
RAM זה מחובר דרך מעגל הפענוח הבא.



ציור 2-30 מעגל פענוח ל-RAM ב-DSM-2200

המפענח 74LS138 מפענח את הקוים A13-A15 וקו IO/M'. ב-8051 עצמו אין קו IO/M', אבל הערכה S-2051 יוצרת קו כזה. הסיבה לכך וכיצד מוסברים בפרק הבא. יציאותיו Y7 ו-Y3 מחוברים דרך גשר בורר אל קו CS' של ה-RAM. הזזת הגשר משנה את מיפוי ה-RAM. לדוגמא, קביעת הגשר מול Y7 קובעת את ה-RAM לתחום הכתובות E000-FFFF, קביעת הגשר מול Y6 קובעת את ה-RAM לתחום הכתובות C000-DFFF וכך הלאה. Y0-Y2 לא ניתנים לגישור כדי לא ליצור חפיפה עם הזכרונות שבערכות S-2051 או S-2085.

הקו RD* הוא קו RD' כאשר משתמשים ב-S-2085. כאשר משתמשים ב-S-2051, זהו קו הנוצר מפעולת AND בין קו PSEN' וקו RD'.

מעבר לבורר תחום הכתובות, נמצאת מערכת גשרים נוספת ליד ה-RAM והמסומנת ב-TS1-TS6. גשרים אלה נועדו על מנת ליזום תקלות בכדי לתרגל איתור תקלות בחומרה. בשלב ראשון ודא שכל הגשרים נמצאים בחלק הנמוך של המעגל (קרוב לשפת הכרטיס).

2.13.1 קבע את הגשר מול Y6 כך שה-RAM יקבע בתחום הכתובות C000-DFFF.

2.13.2 פנה לכתובת C000 והכנס לשם את הנתון 10. המשך לכתובת הבאה C001 והכנס לשם את המספר 11. המשך כך עד לתא C009 והכנס את המספרים 12-19.

2.13.3 חזור ובדוק את התאים C000-C009 ובדוק אם נמצאים שם המספרים 10-19.

2.13.4 הזז את הגשר כך שיהיה מול Y5. בדוק שוב את תוכן התאים C000-C009.

2.13.5 בדוק את תוכן התאים בתחום A000-A009. הסק מסקנות.

2.14 זיהוי דיאגרמת זמנים

2.14.1 קבע את הגשר, כך שיהיה מול Y6, לשם קביעת ה-RAM בתחום הכתובות C000-DFFF.

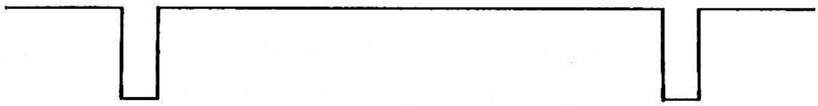
2.14.2 הכנס לתא C3AA את המספר 55H ובדוק שאכן נתון זה נמצא בתא זה.

2.14.3 הרץ את התוכנית הבאה, הקוראת בלולאה אינסופית את הנתון מתא C3AA.

8085	8051
LOOP: LDA C3AA	LOOP: MOV DPTR,#C3AA
STA C3AA	MOVX A,@DPTR
JMP LOOP	MOVX @DPTR,A
	SJMP LOOP

21.4.4 הפעל סקופ כך שיעבוד על מצב DUAL שבו תראה שני ערוצים במסך CH1 ו-CH2.

21.4.5 חבר את הבחון של CH1 לקו 'CS של ה-RAM באופן קבוע וכוון את הסקופ להסתנכרנות על אות זה. הינך צריך לראות אות מהצורה הבאה:



הזמן בין שני האותות הוא זמן מחזור לולאה התוכנית. בכל לולאה פונה ה-CPU פעם אחת לכתובת זו.

2.14.6 חבר את הבחון של CH2 לקו 'OE של ה-RAM. לקו זה מגיע אות 'RD או *RD. כל ירידה ל-"0" של קו זה מסמנת הבאת נתון או הוראה של ה-CPU. שנה את כיול ציר X (ציר הזמן) כך שתזהה בדייקנות את הנקודות בהן קו 'RD משנה מצב מ-Low ל-High. שים לב שקיים מחזור אחד שבו קו 'RD רחב יותר. חשוב מדוע?

2.14.7 עבור עם הבחון של CH2 על רגלי המפענח ורגל ה-RAM, כאשר CH1 מחובר כל הזמן ל-'CS ואינך משנה את הכיול וצייר דיאגרמת זמנים מלאה, הכוללת את אות ה-'CS, קו 'RD (או *RD) ומתחתיהם את הקוים 'IO/M, 'WR, A0-A15, D0-D7.

2.14.8 פענח את תוכן קוי הכתובת וקוי הנתונים עבור כל אחד ממחזורי המכונה. בדוק והסבר אם זה מתאים לתוכנית שהרצת.

2.15 איתור תקלות בחומרה

2.15.1 בדוק שהגשר לקביעת תחום הכתובות מחובר ל-Y6 וגשרי התקלות היזומות נמצאים במקומם.

2.15.2 הזז גשר TS5 למצב תקלה (החלף מצב וקרב אותו לכיוון ה-RAM), ונסה לאתר תקלה זו.

2.15.3 ראשית נבדוק את קוי הבקרה לרכיב. הרץ את התוכנית המתוארת בסעיף 2.14.3).

2.15.4 בדוק ברגלי ה-RAM אם מופיעים אותות 'CS, 'RD ו-'WR. סביר להניח שכן וזה סימן שוקים A13-A15, 'RD, 'WR ו-'IO/M מחוברים הראוי. אם לא, סימן שמעגל DSM-2200 לא תקין.

2.15.5 לחץ RST. נבדוק עתה את קוי הנתונים.

2.15.6 רשום תוכנית הכותבת בתאים C000-C007 את המספרים 01,02,04,08,10,20,40,80 בהתאמה, וחוזרת למוניטור הראשי בסיום הכתיבה. הרץ את התוכנית. אינך יכול להכניס את הנתונים בעזרת תוכנית המוניטור הבסיסי, מכיון שהתוכנית הזו, קוראת את הנתון שוב לאחר כתיבתו. במקרה של אי התאמה מודיעה ERROR.

2.15.7 קרא עתה את תוכן התאים. בגלל שהתקלה יזומה, תמצא בתאים את המספרים הבאים: 01,02,00,00,10,20,40,80. המסקנה היא שקו D2 מקוצר ל-D3.

2.15.8 נתק את הכרטיס DSM-2200 מהערכה S-2100 ובדוק בעזרת בודק הרציפות אם אכן קוים אלה מקוצרים ביניהם.

2.15.9 אם היינו מוצאים רק בתא אחד 00, סימן שהסיבית הרלוונטית מקוצרת לאדמה.

2.15.10 אם היינו מוצאים שסיבית אחת מאולצת ל-"1" בכל התאים, היינו יודעים שהיא מנותקת או מקוצרת ל-VCC.

2.15.11 החזר גשר התקלה TS5 למקומו התקין והחלף מצב לגשר TS1.

2.15.12 חזור על סעיפים 2.15.3 עד 2.15.10 לבדיקת תקינות הקוים 'CS','RD','WR וקוי ה-DATA. סביר להניח שתמצא שהם תקינים.

2.15.13 כבדוק את הקוים A0-A7 ואם יש מיפוי כפול בתחום זה. רשום והרץ את התוכנית הבאה, הממלאה את התאים C000-C0FF במספרים 00-FF, כמוסבר בסעיף 2.10.

8085		8051	
	LXI H,C000		MOV DPTR,#C000
	MVI A,00		CLR A
LOOP:	MOV M,A	LOOP:	MOVX @DPTR,A
	INX H		INC DPTR
	INR A		INC A
	CPI A,FF		CJNE A,#FF,LOOP
	JNZ LOOP		LJMP 0000
	RST 1		

2.15.14 אם יש מיפוי כפול בתחום C000-C0FF הוא יהיה בחזקות של 2 . נקרא את התאים הבאים ונשווה עם מה שהיה צריך להיות לפי התוכנית.

	צריך להיות	נמצא בפועל
C000	00	08
C001	01	01
C002	02	02
C004	04	04
C008	08	08
C010	10	10
C020	20	20
C040	40	40
C080	80	80

אנו רואים שקיים מיפוי כפול בין הכתובות C000 ו-C008 וזה מציין שקו A3 בעייתי. קו זה יכול להיות מנותק, מקוצר לאדמה, או מקוצר ל-VCC. תיתכן כל אחת מהאפשרויות. הסבר מדוע.
נתק את הכרטיס מהערכה ובדוק בעזרת בודק הרציפות את התקלה.

2.15.15 אם היה קצר בין שני קוי כתובת, לדוגמא A2 ו-A3, מה היינו מקבלים לדעתך? זכרו ש-"0" דומיננטי יותר מ-"1". זכור גם את סדר הכתיבה בתאים.

2.15.16 כאשר משתמשים ב-Debugger, רואים את תמונת הבלוק כולו וזה מקל אל איתור התקלה.

2.15.17 תיתכן גם תקלה מסוג של קצר בין קו כתובת לקו נתונים או תקלה בקוי הכתובת הגבוהים.
רשום את פעולותיך לאיתור תקלה בקוי הכתובת הגבוהים A8-A12.
א) הנח תקלה ש-A11 מנותק.
ב) הנח תקלה ש-A11 מקוצר ל-A12.

2.15.18 החזר גשר TS1 למקומו.

2.15.19 חזור על איתור התקלות עבור שאר גשרי התקלות היזומות, ואתר כל תקלה בנפרד. רשום דו"ח על כל גשר והתקלה הנוצרת בגללו.

פורטים

3.1 הסבר כללי ודיאגרמות זמנים

במושג פורט מתכוונים ליחידה המשמשת את ה-CPU להתחברות אל העולם החיצון של מערכת המיקרו-מחשב. פורט מבוא קולט מספר בינארי מהעולם החיצון ומעבירו ל-CPU. פורט מוצא קולט מספר בינארי מה-CPU ומעבירו לעולם החיצון. בדרך כלל, מכיל פורט מוצא תאי Latch או תאי .F.F, השומרים את המספר הבינארי ביציאותיו, עד לפניה הבאה של ה-CPU אליו. הפורטים נקראים גם יחידות I/O (Input/Output).

ל-8085 יש שתי פקודות - OUT ו-IN שבזמן השימוש בהם עולה קו 'IO/M ל-High, ובקוי הכתובת A0-A7 מופיעה הכתובת של הפורט המבוקש. מעגל הפענוח, במקרה זה, צריך לפענח את עליית קו 'IO/M ל-High ומחצית מקוי הכתובת לשם יצירת 'CS לפורט.

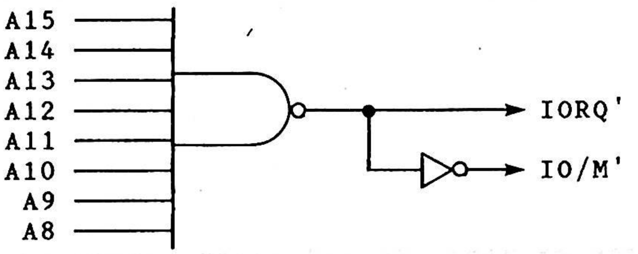
בפניה ליחידות I/O ב-8085, גם הקוים A8-A15 מקבלים את כתובת הפורט המבוקש, במקביל לקוים A0-A7, כך שניתן לפענח את הקוים A8-A15 במקום A0-A7. מעגל הפענוח, בנוסף לפענוח קו 'IO/M ומחצית קוי הכתובת, צריך לפענח את הקוים 'RD או 'WR. אם הפורט הוא פורט מבוא - את קו 'RD, היורד ל-Low בפקודה IN. אם הפורט הוא פורט מוצא, הוא מפענח את קו 'WR, היורד ל-Low בפקודה OUT.

ה-8088 דומה ל-8085. גם כאן קיימות פקודות IN ו-OUT. בפקודות אלה קו 'IO/M עולה ל-High, אבל הכתובת המצוינת

היא כתובת בת 16 קוים (A0-A15). כלומר, תחום ה-I/O הוא תחום בן 64K כתובות.

ל-8051 אין קו 'IO/M ואין פקודות מיוחדות לפניה ליחידות I/O. קובעים את פורטי המבוא והמוצא במיפוי זכרון. מעגל הפענוח מפענח את הקוים A0-A15 ואת הקוים 'RD ו-'WR בהתאם לסוג הפורט (מבוא או מוצא). הפניה לפורטים נעשית כפניה לתא בזכרון RAM חיצוני בדרך כלל.

בוחרים תחום כתובות כתחום I/O. לדוגמא, בערכה S-2051 נבחר התחום FF00-FFFF כאיזור ה-I/O. במקרה כזה יוצרים קו 'IORQ (I/O ReQuest) היורד ל-Low כאשר הוא מזהה פניה לתחום זה. למשל המעגל הבא:



ציור 3-1 יצירת קו 'IO/M ל-8051

חיבור קו זה למהפך, יוצר לנו מעין קו 'IO/M. נתייחס לקו זה בכל מעגלי הפענוח הבאים.

גם ב-8085 וב-8088 ניתן לבנות יחידות I/O גם במיפוי זכרון, אם רוצים. למיפוי זכרון יש יתרון בזה שמספר הפקודות לפניה לתא זכרון רב יותר מאשר בפניה ליחידות I/O.

גם ב-APPLIC-1 (הכרטיס הנתקע ב-PC) נעשה פענוח ראשוני. במחשבי IBM מיועד תחום כתובות ה-I/O 0300H-03FFH לכרטיסי הרחבה (בקר למדפסת, בקר לצג, תקשורת טורית

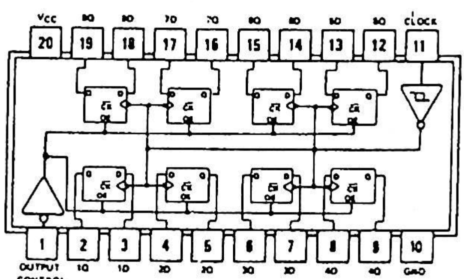
RS232 וכד'). בתוך תחום זה, התחום 0310H-031FH והתחום 0350H-035FH מיועדים לכרטיסי משתמשים מיוחדים. ב-APPLIC-1 נעשה פענוח לשני תחומים אלה, וכאשר מתבצעת במחשב פניה לכתובת I/O באחד התחומים, יורד קו 'IORQ ל-Low. גם כאן חיבור מהפך לקו זה יצור לנו מעין קו 'IO/M.

כאשר נרצה לציין מעתה פניה לכתובת I/O כלשהי, נציין זאת בעזרת הביטוי IO-xx. לדוגמא הכתובת IO-53H משמעותה כתובת I/O 53H ב-8085, כתובת FF53H ב-8051 וכתובת 0353H ב-8088.

קיימות יחידות הכוללות בתוכן מספר פורטים ולכן תופסות מספר כתובות.

3.2 פורט מוצא פשוט

כ-פורט מוצא נשתמש ברכיב 74LS374. רכיב זה מורכב משמונה תאי .F.F בני סיבית אחת, בעלי כניסת Clock משותפת כמתואר בציור 3-2.

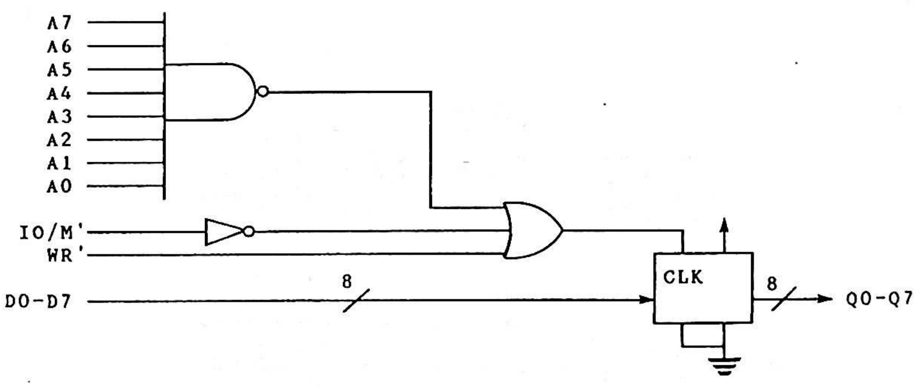


ציור 3-2 תיאור הרכיב 74LS374

ליציאות התאים יש חוצצים ולהם קו 'OE (Output Enable).
אם רוצים שתוכן התאים יופיע באופן קבוע ביציאות הרכיב, נחבר את קו 'OE (רגל 1) באופן קבוע ל-GND. את קוי הנתונים (D0-D7) נחבר לקוים של מסלול הנתונים של ה-CPU.
יחידת פלט זו היא יחידה בת תא אחד בלבד ולכן תתפוס כתובת אחת. לשם דוגמא, נקבע פורט זה IO-FF. מעגל הפענוח צריך להוריד את קו CK של ה-74LS374 (רגל 11) ל-Low, כאשר כותב ה-CPU בכתובת זו. המידע שעל קוי הנתונים יעבור לתאי הרכיב, כאשר עולה קו זה ל-High. המידע שנמסר ישאר נעול בתאי ה-.F.F עד לכתיבה הבאה בכתובת זו.

מעגל הפענוח חייב לפענח גם את קוי הבקרה 'IO/M ו-'WR.

מעגל הפענוח יראה כך:



ציור 3-3 מעגל פענוח ל-פורט מוצא

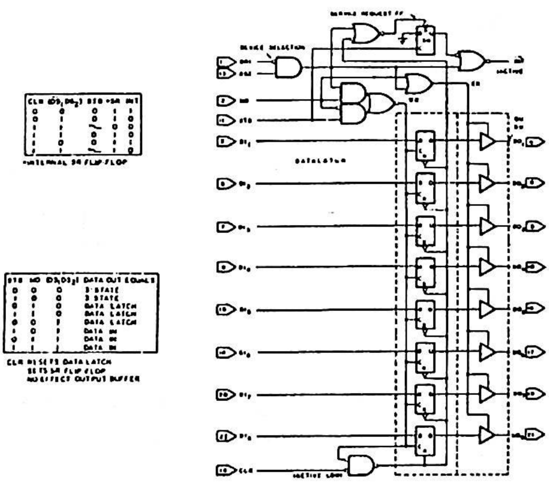
ישנם מספר רכיבים נוספים הדומים ל-74LS374 ותואמים לו מבחינת מספרי רגליים. אחד מהם הוא ה-74LS373. רכיב זה מכיל 8 תאי Latch ולא תאי .F.F. כדי שהמידע יכנס אליו, צריכה כניסת ה-CLK שלו לעלות ל-1. נעילת המידע נעשית עם ירידת קו ה-CLK ל-Low. ההבדל בין ה-Latch לבין ה-.F.F הוא, שב-.F.F המידע נכנס רק בזמן מעבר של ה-CLK מ-"0"

ל-"1", וב-Latch הוא נכנס ומופיע ביציאות בכל זמן שה-CLK נמצא ב-High.

עם עליית קו ה-CLK ל-High ב-74LS373, המידע בקוי הכתובים עדיין אינו יציב, ויתכן שיופיע מידע שגוי ביציאותיו. עם נעילת המידע, יהיה המידע יציב ונכון. זו הסיבה שעדיף להשתמש ב-74LS374.

רכיב אחר הדומה ל-74LS374 הוא הרכיב 74LS273. רכיב זה אינו כולל קו 'OE ויציאות תאי ה-F.F. מחוברות ליציאות הרכיב באופן קבוע. במקום קו 'OE, כולל הרכיב קו 'CLR. הורדת קו זה ל-Low, מאפסת את תאי ה-F.F..

פורט מוצא ניתן לבנות גם בעזרת רכיב חכם מעט יותר היוצר בקשת פסיקה. רכיב כזה הוא הרכיב 8212, המתואר בציור הבא:

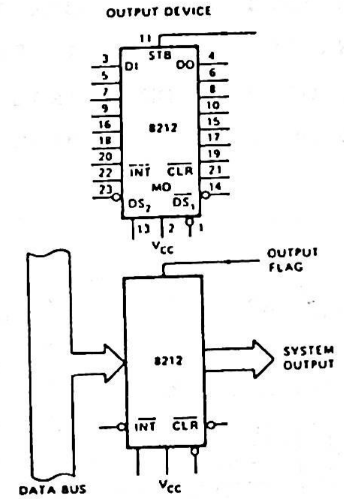


ציור 3-4 תיאור ה-8212

ה-8212 כולל תאי Latch שיציאותיהם נשלטות על-ידי חוצצים. לרכיב 2 קוי 'CS (DS1 ו-DS2). מלבד מרכיבים אלה המאפיינים פורטים יש לו קוים נוספים. קו CLR הגורם

לאיפוס תוכן תאי ה-Latch. קו MD הקובע אם הוא עובד ב-MODE OUT (MD=VCC) או ב-MODE IN (MD=0).

הקוים המאפשרים ביצוע פסיקה הם הקוים STB ו-'INT. את קו 'INT מחברים לקו פסיקה של ה-CPU. אם קו הפסיקה מופעל ב-High, מחברים אותו דרך מהפך. הורדת קו STB מ-High ל-Low, גורמת לבקשת פסיקה בקו 'INT.



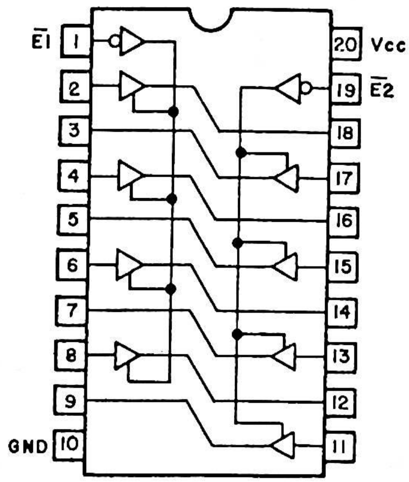
ציור 3-5 חיבור ה-8212 כפורט מוצא

בציור מתואר אופן חיבור ה-8212 כפורט מוצא. ירידת קו STB ל-Low, יוצרת בקשת פסיקה, אבל לא מכניסה דבר לתאי ה-Latch (MD=1). בתוכנית הפסיקה, מוסר ה-CPU נתון לפורט. אז המערכת החיצונית, יוזמת את ה-STB, אוספת אליה את הנתון. קו STB מחובר בכל מקרה למערכת החיצונית ל-CPU.

3.3 פורט מבוא פשוט

כפורט מבוא פשוט נשתמש ברכיב 74LS244. זהו, למעשה, חוצץ בלבד. הוא איננו אוגר את נתוני הכניסה. הרכיב רק מפריד בין קוי הנתונים של מערכת המיקרו-מחשב לעולם החיצון. הוא נפתח רק כאשר ה-CPU פונה אליו לשם קריאת נתון ממערכת חיצונית המחוברת לכניסותיו.

הרכיב בנוי מ-8 חוצצים הערוכים בשתי רביעיות, כאשר לכל רביעיה קו Enable נפרד, כמתואר בציור 3-6. הורדת קו ה-Enable ל-Low, מוציאה את החוצצים מ-3-STATE ומאפשרת מעבר המידע מקוי הכניסה לקוי היציאה של החוצצים.

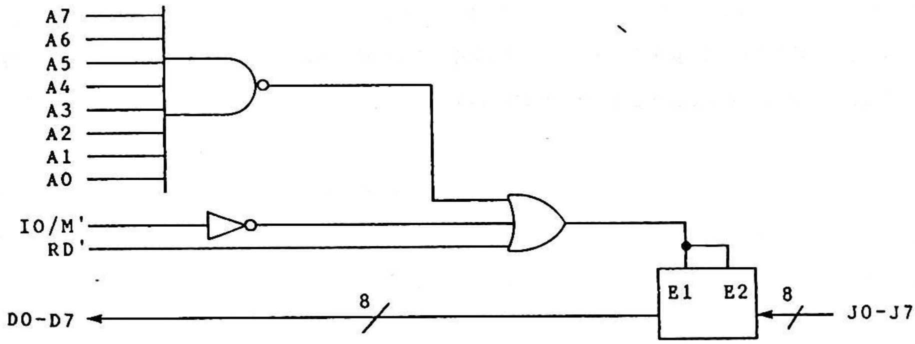


ציור 3-6 תיאור ה-74LS244

גם יחידת קלט זו תופסת כתובת של תא אחד בזכרון. בדוגמא, נקבע את כתובתה של יחידת הפלט הקודמת (IO-FF) בעזרת מעגל פענוח.

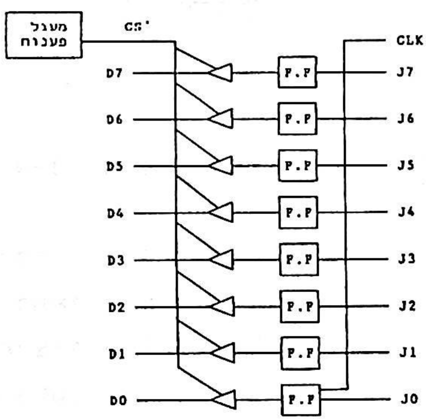
יציאת ה-'CS של מעגל פענוח זה מתחבר לרגליים 1 ו-19 של ה-74LS244. קו זה צריך לרדת ל-Low כאשר קורא ה-CPU מכתובת זו. מעגל הפענוח חייב לפענח גם את קוי הבקרה 'IO/M ו-'RD.

מעגל הפענוח יראה כך:



ציור 3-7 מעגל פענוח ל-פורט מבוא

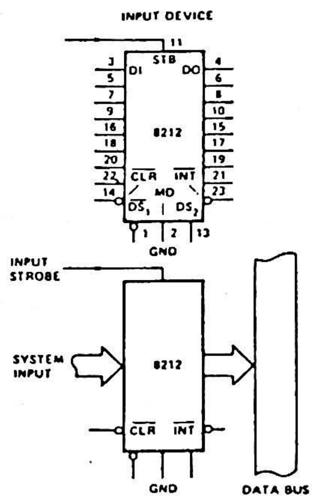
ניתן לבנות פורט מבוא הכולל תאי זכרון (Latch או .F.F). זאת בעזרת הרכיבים שתוארו קודם בבניית פורט מוצא. לדוגמא, שימוש ברכיב 74LS374 כפורט מבוא.



ציור 3-8 74LS374 כפורט מבוא

מערכת מהסוג הזה מיועדת לאיסוף נתונים מהירים. המערכת החיצונית, כאשר יש לה נתון למסירה, מוסרת את הנתון ויוצרת CLK. הנתון ננעל בתאי ה-.F.F אבל איננו יוצא למסלול הנתונים. ה-CPU בזמנו החופשי, פונה אל הרכיב (פניה זו פותחת את החוצצים במוצא), ואוסף את הנתון. יש לדאוג, כמובן, שה-CPU ידע שנתון נמסר לפורט המבוא, באמצעות פסיקה, בדרך כלל.

רכיב העושה כל זאת בצורה טובה יותר הוא ה-8212 שתואר קודם. לרכיב זה יש MODE עבודה כפורט מבוא, המתואר בציור הבא.



ציור 9-3 חיבור ה-8212 כפורט מבוא

במקרה זה, ירידת קו STB ל-Low גורמת לנעילת המידע שבקוי הכניסה בתאי ה-Latch ולבקשת פסיקה. את המידע מספקת מערכת חיצונית (Input Device). ה-CPU בתוכנית הפסיקה פונה לפורט, אוסף את הנתון, מעבד אותו במידת הצורך ויכול לחזור ולהמשיך בתוכנית הראשית.

שיטה זו משחררת את ה-CPU מהצורך לדגום את פורט המבוא, כדי לזהות נתון נכנס.

3.4 פורטי הערכה

הערכה S-2100 כוללת ארבעה פורטים בנויים בתוכה. יציאות וכניסות הפורטים נמצאות על הפנל. הם מיועדים לתרגול יישומי פורטים.

2 פורטי המבוא בערכה בנויים מרכיבי 74LS244. אחד נקרא INPUT PORT A ונמצא בכתובת IO-11H. השני נקרא INPUT PORT C ונמצא בכתובת IO-12H.

אחד מפורטי המוצא הבנויים בערכה הוא הרכיב 74LS374. הוא מכונה OUTPUT PORT B, וקבוע בכתובת IO-11H.

פורט המוצא השני בערכה בנוי מרכיב 8212 ומכונה OUTPUT PORT D. פורט זה נקבע בכתובת IO-12H.

את המידע האגור ב-8212 ניתן לאפס בעזרת קו CLR.

ה-8212 מאפשר יצירת פסיקה ל-CPU והעברת נתונים באמצעות פסיקה. נושא זה יתואר בהמשך הפרקים.

3.5 פורט הניתן לתכנות PIO-8255

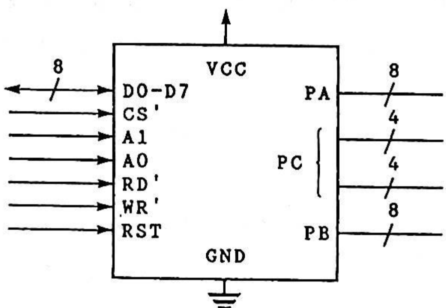
בשם פורט הניתן לתכנות (Programmable Port) מתכוונים לרכיב, שבדרך כלל, יש לו מספר אופני פעולה ואופן הפעולה נקבע בתוכנה. גם קווי ניתנים לקביעה בתוכנה כקווי OUTPUT

או קוי INPUT או שילוב שלהם. רכיבים מהסוג הזה הם הרכיבים 8255, 6821 ועוד. רכיב כזה תופס יותר מכתובת אחת בזכרון. שני הרכיבים הנזכרים, תופסים ארבע כתובות כל אחד. לכן לרכיבים אלה בונים מעגל פענוח לקוים A2-A7 ואת יציאתו מחברים לקו 'CS. את הקוים A0,A1 מחברים ישירות לרכיב על מנת לאפשר בחירה בין הכתובות השונות.

קיימים רכיבים הניתנים לתכנות ומיועדים למטרות מוגדרות. לדוגמא, לתקשורת טורית, בקר ללוח מקשים, בקר לצג, בקר לכונן דיסקטים ועוד. הפניה לרכיבים אלה היא בדרך כלל כפניה לפורטים. נכיר מספר רכיבים כאלה בהמשך.

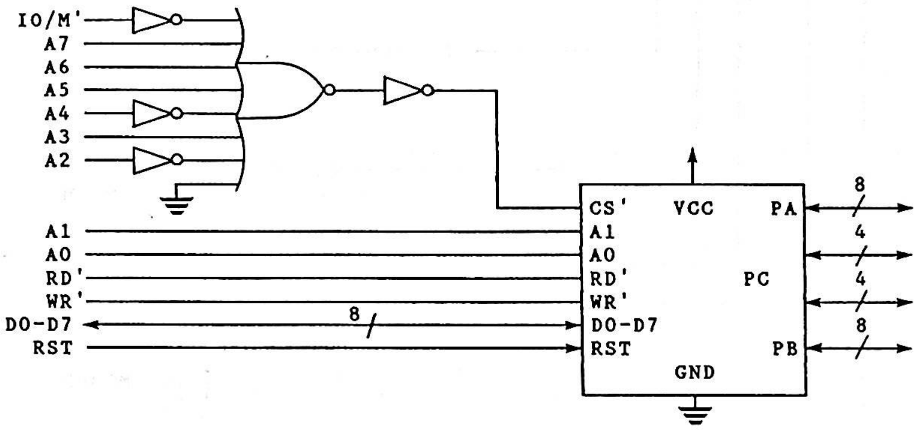
ב-8255, נמצא באחת הכתובות (A0,A1=11) אוגר בקרה. פניה לאוגר זה והכנסת מספר מתאים קובעת את אופני פעולתו של הרכיב ויעוד פורטיו (IN או OUT). לרכיב שלושה פורטים - פורט A, פורט B ופורט C. סה"כ 24 קוי I/O.

ה-8255 מתואר בציור הבא:



ציור 3-10 תיאור ה-8255

בכוונה מוקם פורט C בין פורט A ופורט B, בגלל תפקודו המיוחד כפי שנראה בהמשך. ראשית נראה כיצד לחבר את הרכיב אל ה-CPU. נניח שרוצים לחברו לכתובות IO-14 עד IO-17 (כאמור הוא תופס ארבע כתובות). מעגל הפענוח יראה בצורה הבאה:

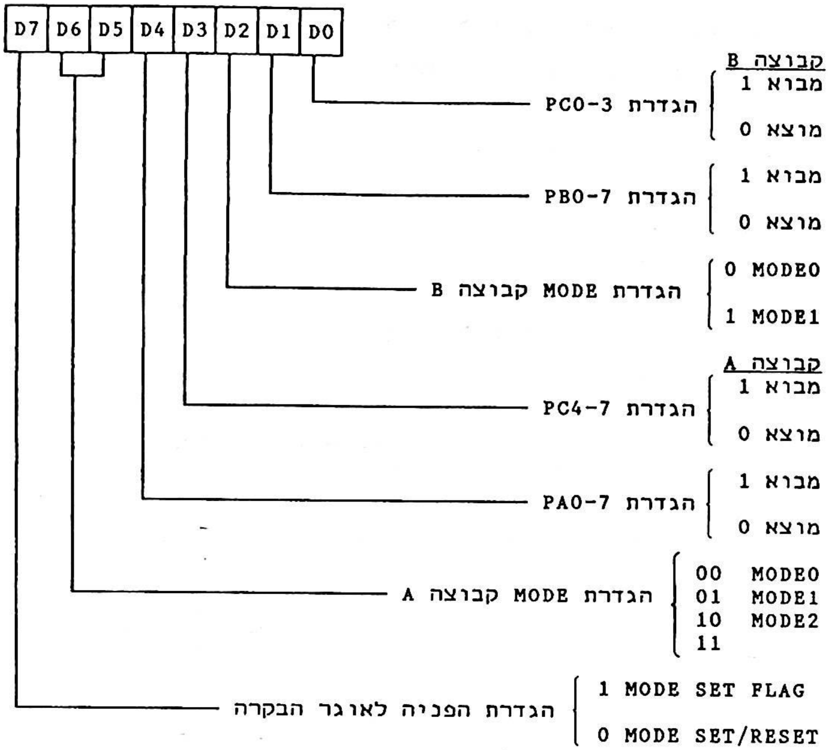


ציור 3-11 חיבור ה-8255 ל-CPU לכתובות IO14-IO17

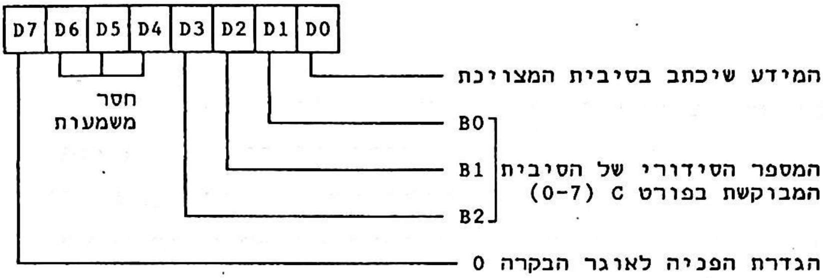
בפניה לכתובת IO-17 (A1,A0=11), פונה ה-CPU לאוגר הבקרה של הרכיב. המלה הנכתבת באוגר הבקרה קובעת את אופן תפקוד הרכיב. ניתן לשנות את תפקודו. לדוגמא, לקבוע בתוכנית את PA פעם אחת כפורט מבוא ובפעם אחרת כפורט מוצא.

הפורטים מחולקים לשתי קבוצות. קבוצה A הכוללת את פורט A ו-4 הסיביות הגבוהות (PC7-PC4) של פורט C. קבוצה B כוללת את פורט B ו-4 הסיביות הנמוכות (PC3-PC0) של פורט C. לקבוצה B יש שני אופני עבודה (MODE0 ו-MODE1) ולקבוצה A יש שלושה אופני עבודה (MODE0, MODE1 ו-MODE2).

מלת הבקרה הנכתבת לאוגר הבקרה קובעת את אופן הפעולה של הרכיב בצורה הבאה:



כדי שאוגר הבקרה יקלוט את אופן הפעולה הנדרש בהתאם לסיביות D0-D6, צריכה להיות הסיבית D7, בפניה אליו, להיות ב-"1". זה מציין שהמלה הנכתבת היא מלה לקביעת תפקוד הרכיב. כאשר ב-D7 יש "0" בכתיבה לאוגר הבקרה, תוכן אוגר הבקרה לא משתנה מתוכנו הקודם. במקום זה מאולצת ל-"1" (SET) או מאולצת ל-"0" (RESET) אחת מסיביות פורט C. כלומר, פורט C ניתן להפעלה במיעון סיבית. במקרה כזה תפקוד המלה הנכתבת בכתובת הגבוהה (A1,A0=11) היא כדלקמן.



0 - MODE SET/RESET

MODE0

ב-MODE0 מתפקדים הפורטים כפורטים פשוטים. פורט מבוא או פורט מוצא. קיימות 16 אפשרויות. פורט C מחולק לשניים וכל חלק יכול לתפקד כפורט מבוא או כפורט מוצא.

קבוצה B		קבוצה A		מלת בקרה בהקסא.	מלת בקרה בבינארית
פורט B	פורט C (LSB)	פורט C (MSB)	פורט A		
					D- 76543210
מוצא	מוצא	מוצא	מוצא	80	10000000
מוצא	מבוא	מוצא	מוצא	81	10000001
מבוא	מוצא	מוצא	מוצא	82	10000010
מבוא	מבוא	מוצא	מוצא	83	10000011
מוצא	מוצא	מבוא	מוצא	88	10001000
מוצא	מבוא	מבוא	מוצא	89	10001001
מבוא	מוצא	מבוא	מוצא	8A	10001010
מבוא	מבוא	מבוא	מוצא	8B	10001011
מוצא	מוצא	מוצא	מבוא	90	10010000
מוצא	מבוא	מוצא	מבוא	91	10010001
מבוא	מוצא	מוצא	מבוא	92	10010010
מבוא	מבוא	מוצא	מבוא	93	10010011
מוצא	מוצא	מבוא	מבוא	98	10011000
מוצא	מבוא	מבוא	מבוא	99	10011001
מבוא	מוצא	מבוא	מבוא	9A	10011010
מבוא	מבוא	מבוא	מבוא	9B	10011011

MODE1

ב-MODE0 מתפקדים הפורטים של ה-8255 כפורטים פשוטים - פורטי מוצא או פורטי מבוא. בפניה לפורט פשוט אין התייחסות למצב היחידה המחוברת לפורט לפני פניית ה-CPU אליה. כאשר קורא ה-CPU נתון מפורט המבוא, אין הוא יכול לדעת האם היחידה החיצונית המחוברת לפורט המבוא מוכנה עם הנתון. אם ה-CPU כותב נתון בפורט מוצא, אין הוא יכול לדעת אם היחידה החיצונית מוכנה לקבל נתון. כאשר קוראים מפסק או מדליקים LED, אין זה חשוב. אבל, כאשר רוצים לבצע תקשורת מקבילית בין המיקרו-מחשב למערכת חיצונית, מדפסת למשל, זה חשוב.

נתייחס בנפרד להתנהגות המיקרו-מחשב בפורט מבוא ובפורט מוצא. נניח שיחידה חיצונית מחוברת לפורט מבוא של המחשב. היחידה החיצונית למחשב רוצה למסור נתון לפורט מבוא של המחשב ולא להמתין עד שהוא יאסוף אותה. מצד שני, אין רוצים שהמחשב ידגום את פורט המבוא כל הזמן כדי שיזהה פניה של היחידה החיצונית אליו. במקרה כזה משתמשים בפורט מבוא מתוחכם, כמו ה-8212 או ה-8255 בצורה מתוחכמת יותר. ב-8255 מדובר בעבודה ב-MODE1. כאשר מגדירים את אחד הפורטים A או B לשמש כפורט מבוא ב-MODE1 (ראה בתיאור אוגר הבקרה כיצד), משמשים שלושה קוים בפורט C כקוי בקרה מיוחדים עבורו. אותות הבקרה המיוחדים לפורט מבוא כזה הם האותות הבאים:

STB (Strobe) - אות מבוא

היחידה החיצונית מכינה בכניסות פורט המבוא של המחשב את הנתון המיועד למסירה ומורידה את כניסת STB ל-"0". הורדת אות מבוא זה ל-"0" גורמת לנעילת המידע בכניסות פורט המבוא בתוך פורט המבוא. פורט המבוא, במקרה זה, כולל תאי F.F. וקו זה משמש עבורו כקו CLK.

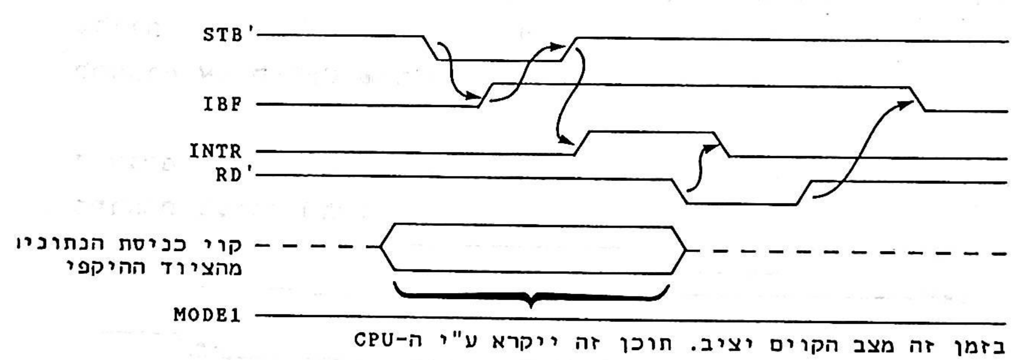
IBF (Input Buffer Full) - אות מוצא

עם נעילת המידע בפורט המבוא עולה קו זה ל-"1". הוא מציין ליחידה המוסרת שהמידע ננעל. ירידתו ל-"0" שוב, מציינת ליחידה המוסרת שהנתון נאסף על-ידי ה-CPU והיא יכולה למסור נתון נוסף. הירידה ל-"0" נעשית כאשר ה-CPU פונה לפורט המבוא וקורא את הנתון.

INTR (INTeRrupt) - אות מוצא

אות זה ניתן לחבר לכניסת פסיקה של המחשב הקולט. הוא עולה ל-"1", לאחר שקו STB ירד ל-"0" ועלה ל-"1" שוב. בתוכנית הפסיקה יפנה ה-CPU לפורט המבוא ויאסוף את הנתון. בצורה כזו, פונה ה-CPU לפורט המבוא רק כאשר מחכה לו שם נתון חדש. פורט המבוא מנהל בצורה עצמאית את כל התקשורת עם היחידה המוסרת.

דיאגרמת הזמנים המתארת את הקשר בין האותות השונים מתוארת בציור הבא:



ציור 3-12 אותות הבקרה בפורט מבוא

גם בפורט מוצא, ניתן לתת לפורט לנהל את התקשורת עם היחידה קולטת. גם כאן, כאשר מגדירים את פורט A או פורט B לשמש כפורט מוצא ב-MODE1 משמשים שלושה קוים של פורט C כקוי בקרה עבור פורט המוצא. אותות הבקרה המיוחדים לפורט מוצא כזה הן האותות הבאים:

<u>OBF (Output Buffer Full) - אות מוצא</u>

ירידת קו זה ל-"0" מסמנת ליחידה הקולטת שה-CPU המשדר כותב מידע בפורט המוצא ומידע זה ממתין לאיסופו.

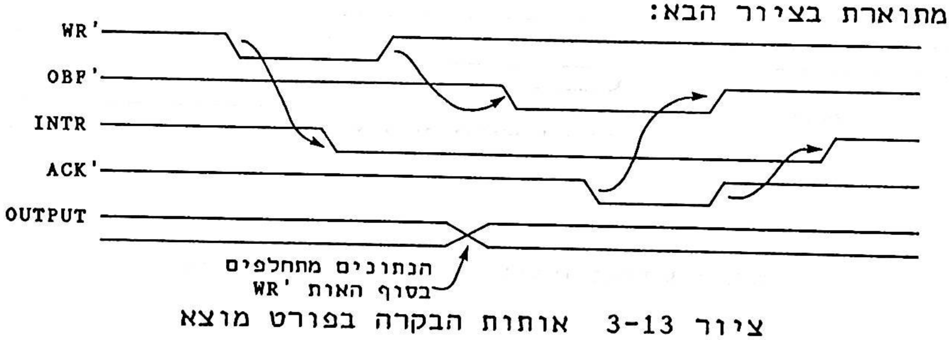
<u>ACK (ACKnowledge) - אות מבוא</u>

קו זה משמש ליחידה הקולטת להודיע למשדר על קליטת הנתונים. המחשב הקולט מוריד קו זה ל-"0" ומעלה אותו ל-"1" לשם כך.

<u>INTR (INTeRrupt) - אות מוצא</u>

קו זה ניתן לחבר לכניסת פסיקה של ה-CPU המשדר. קו זה ירד ל-"0" עם כתיבת הנתון על-ידיו, ויעלה ל-"1" לאחר עליית קו ה-ACK ל-"1". עליה זו של קו ה- INTR יכולה להפכות את ה-CPU שוב לשגרת פסיקה שתשדר מידע נוסף.

דיאגרמת הזמנים המתארת את הקשר בין האותות השונים מתוארת בציור הבא:



ציור 3-13 אותות הבקרה בפורט מוצא

כיתן לחסום או לאפשר את בקשות הפסיקה ב-8255 עצמו. כאשר משתמשים בסיביות פורט C כקוי כניסה, ניתן לאלץ ב-.F.F הפנימי שלהם "1" או "0" (SET או RESET) ב-SET/RESET MODE של אוגר הבקרה, המתואר ב-MODE0. במקרה כזה נקראת הסיבית בשם INTE, שמשמעותה INTerrupt Enable. כאשר קו זה עבור הפסיקה המתאימה הוא ב-"1", מאופשרת הפסיקה, וחסומה כאשר קו זה ב-"0".

סכום תפקוד סיביות פורט C ב-MODE1 מופיע בטבלה הבאה:

מלת בקרה D4 D3 D2 D1 D0	ערוץ A	ערוץ B	ערוץ C							
			PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
11/011X	מבוא	מבוא	I/O	I/O	IBF_A	STB_A $INTE_A$	$INTR_A$	STB_B $INTE_B$	IBF_B	$INTR_B$
01/010X	מוצא	מוצא	OBF_A	ACK_A $INTE_A$	I/O	I/O	$INTR_A$	ACK_B $INTE_B$	OBF_B	$INTR_B$
11/010X	מבוא	מוצא	I/O	I/O	IBF_A	STB_A $INTE_A$	$INTR_A$	ACK_B $INTE_B$	OBF_B	$INTR_B$
01/011X	מוצא	מבוא	OBF_A	ACK_A $INTE_A$	I/O	I/O	$INTR_A$	STB_B $INTE_B$	IBF_B	$INTR_B$

MODE2

MODE2 קיים רק עבור פורט A. ב-MODE עבודה זה משמש פורט A כפורט מבוא ומוצא בו זמנית. מיועד לתקשורת דו כיוונית. גם כאן קוי פורט C משמשים כאותות בקרה, גם

עבור פורט המבוא וגם עבור פורט המוצא. תפקידי הסיביות מתוארות בטבלה הבאה:

PC_7	PC_6	PC_5	PC_4	PC_3	PC_2 ,PC_1 ,PC_0
OBF_A	ACK_A $INTE_1$	IBF_A	STB_A $INTE_2$	INTR	מבוא או מוצא

תפקיד הקוים דומה להגדרתם ב-MODE1, מלבד הבדל אחד. כאשר פונה ה-CPU לפורט A וכותב בו מידע כפורט מוצא, יורד OBF ל-"0" כמו ב-MODE1 אבל המידע לא מופיע ביציאות פורט A. היציאות נמצאות ב-3-STATE, כי ייתכן שהיחידה הקולטת עדיין נמצאת במצב שידור. OBF מסמן לה שיש מידע הממתין לה. עם הורדת קו ACK ל-"0", על-ידי היחידה הקולטת, מופיע המידע ביציאות הפורט ונאסף על-ידה.

סיביות קוי פורט C, המשמשים כאותות מבוא STB (PC4) ו-ACK (PC6), משמשות גם כסיביות איפשור. פסיקה. INTE1 (PC6) לאיפשור פסיקה למקרה של הפעלת הפורט כפורט מוצא ו-INTE2 (PC4) לאיפשור פסיקה למקרה של הפעלת הפורט כפורט מבוא.

מלת הבקרה להעברת פורט A ל-MODE2 היא המלה הבאה:

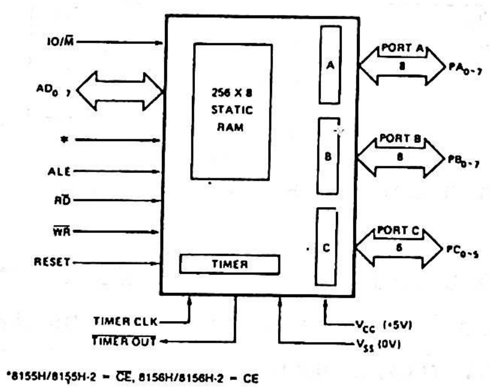
11XXXBBB

הסיביות המסומנות ב-B,B,B מיועדות לקביעת מצבם של פורט B, אופן פעולתו וסיביות PC0-PC2 של פורט C.

XXX מסמן שאין משמעות לסיביות אלה.

3.6 פורט הניתן לתיכנות עם RAM ו-TIMER פנימי - 8155

רכיב זה פופולארי במערכות המבוססות על רכיבי אינטל. למעשה, קיים רכיב תאום נוסף, הנקרא 8156. הוא מכיל 3 פורטים הניתנים לתכנות (שניים בני 8 סיביות ואחד בן 6 סיביות), מונה בן 14 סיביות ו-RAM בן 256 בתים. בנוסף לכך, מיועד הרכיב להתחבר ישירות אל קוי AD0-AD7 וקו ALE. בעזרת קו ה-ALE הוא מסוגל להבחין בין אותות ה-DATA וה-ADDRESS שעל קוים אלה. דיאגרמת המלבנים שלו מתוארת בציור הבא:



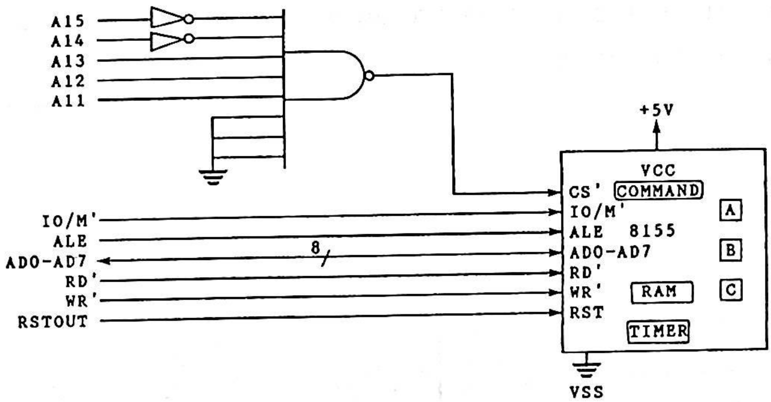
ציור 3-14 תיאור מלבני של ה-8155/8156

בדיאגרמה רואים שלרכיב יש קו *CE (CS). ההבדל היחיד בין ה-8156 ל-8155 הוא בקו זה. ב-8156 קו זה הוא קו CE וב-8155 הוא קו 'CE.

קו ה-CE מתקבל מפענוח הקוים A8-A15. פענוח זה קובע את תחום הכתובות אותו תופס הרכיב בפקודות פניה לזכרון. לרכיב אוגרים פנימיים. לאוגרים אלה ניתן לפנות בפניה

לפורטים (פקודות IN ו-OUT). ב-8085, הכתובות שתופסים אוגרים אלה מתייחסות לתחום הכתובות של RAM הרכיב. זאת מכיוון, שבפקודות I/O הכתובת ב-A0 עד A7 זהה לכתובת ב-A8 עד A15.

לדוגמא, נניח ש-RAM הרכיב נקבע לתחום הכתובות 3800-38FF, אזי יתפסו האוגרים הפנימיים את הכתובות 38-3D. אופן חיבור הרכיב מתואר בציורים הבאים:

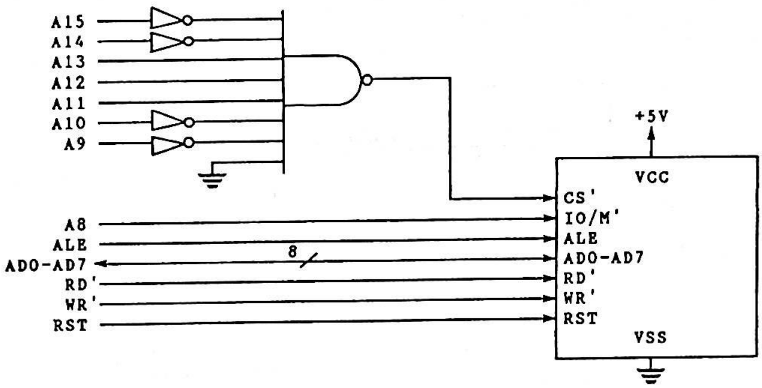


ציור 3-15 אופן חיבור ה-8155 ל-8085

חיבור זה מבוסס על כך שבפניה ליחידת I/O מקבלים קוים A8-A15 את כתובת יחידת ה-I/O, כמו בקוים A0-A7. מכיוון שיש 6 כתובות I/O מתבצע פענוח על הקוים A15-A10 בלבד. כלומר, פניה לכתובות IO-38 עד IO-3F תיצור CS לרכיב. זיהוי לאיזה פורט הכוונה מתבצע בתוך הרכיב בהתאם למספר הרשום בקוים A0-A7. כאמור כולל הרכיב Latch פנימי.

כאשר מתבצעת פניה לכתובת זכרון בתחום 3800-38FF, קו IO/M' ב-"0", וזה מסמן לרכיב שהפניה היא לתא ב-RAM הפנימי. במקרה של פניה לזכרון נוצר כאן מיפוי כפול. אין פענוח של כל הקוים A8-A15 ולכן יתקבל CS גם בפניה לתחומים 3900-39FF, 3A00-3AFF וכך הלאה עד 3F00-3FFF. כלומר, מבחינת זכרון תופס הרכיב את התחום 3800-3FFF (2K תאים)למרות היותו בן 256 בתים בלבד.

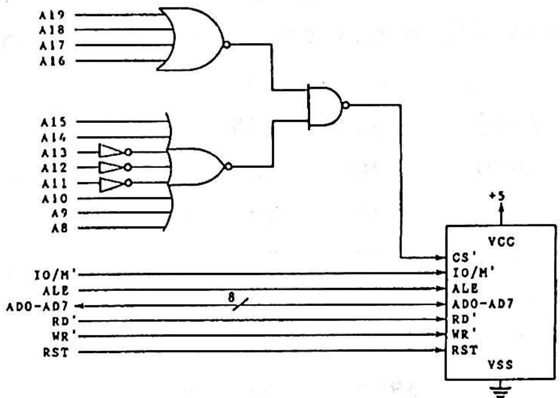
ב- 8051 אין קו 'IO/M ואת יחידות ה-I/O מחברים במיפוי זכרון. במקרה כזה, נקבע את ה-RAM לתחום הכתובות 3800-38FF ואת יחידות ה-I/O לתחום 3900-3905 והלאה.
קו A8 ישמש כקו 'IO/M בצורה הבאה:



ציור 3-16 אופן חיבור 8155 ל-8051

הפניה לאוגר COMMAND תהיה בכתובת 3900H, לפורט A בכתובת 3901H וכך הלאה.

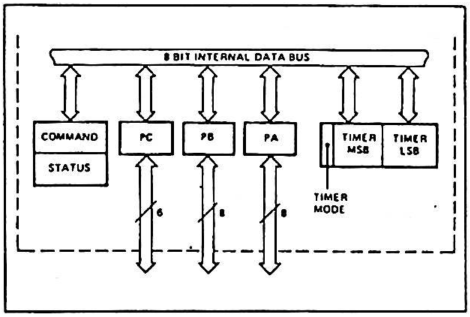
במקרה של ה-8088 נקבע את הרכיב בתחום 03800-038FF. עלינו לקבוע אותו ב-64K הנמוכים, כי הקוים A16-A19, בפניה לכתובת I/O נמצאים ב-"0". מעגל הפענוח יהיה המעגל הבא:



ציור 3-17 אופן חיבור ה-8155 ל-8088

אוגר COMMAND יתפוס את הכתובת IO-3800H, פורט A את הכתובת IO-3801H וכך הלאה.

דיאגרמת המלבנים של האוגרים הפנימיים וקוצב הזמן מתוארת בשרטוט הבא:



ציור 3-18 תיאור האוגרים הפנימיים של ה-8155

ה-8155 תופס 6 כתובות I/O פורט. שלוש כתובות תופסים הפורטים A,B ו-C. כתובת אחת תופס אוגר הנקרא COMMAND/ STATUS ושתי כתובות תופס קוצב הזמן (אחת ל-BYTE הנמוך ואחת ל-BYTE הגבוה של הקוצב).

בדוגמא הקודמת שבה ה-RAM תופס את התחום 3800-38FF, יתפסו האוגרים הפנימיים את כתובות ה-I/O הבאות:

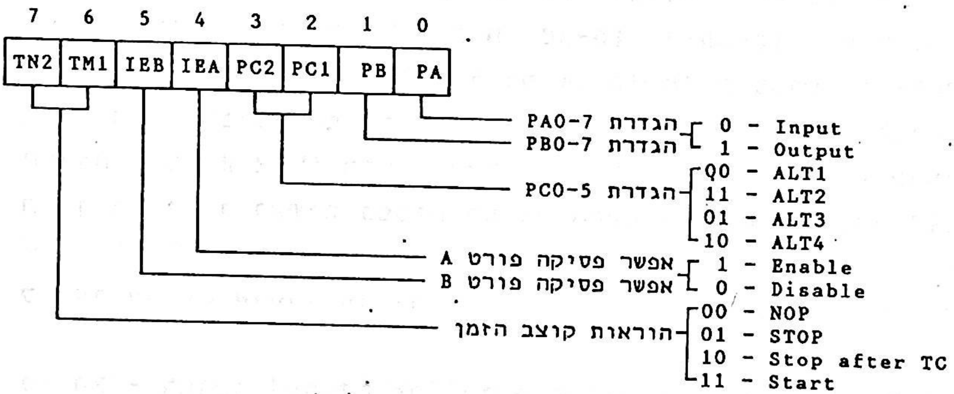
		8085	8051	8088
אוגר COMMAND/STATUS	-	38	3900	3800
פורט A	-	39	3901	3801
פורט B	-	3A	3902	3802
פורט C	-	3B	3903	3803
BYTE נמוך	-	3C	3904	3804
BYTE גבוה	-	3D	3905	3805

+ 2 סיביות MODE

פורטים A ו-B ניתנים לקביעה כפורטי קלט או פלט. לגבי פורט קיימות 4 אפשרויות להפעלתו והן:

PIN	ATL.1	ALT.2	ATL.3	ATL.4
PC0 PC1 PC2 PC3 PC4 PC5	Input port	Output Port	Port A Interrupt Port A Buffer Full Port A Strobe Output Port	Port A Interrupt Port A Buffer Full Port A Strobe Port B Interrupt Port B Buffer Full Port B Strobe

גם לקוצב הזמן יש ארבעה אופני הפעלה. קביעת אופני ההפעלה נעשית בשתי הסיביות העליונות של אוגר ה-MSB של קוצב הזמן. האפשרויות הן: גל ריבועי בודד, גל ריבועי רצוף, פולס בודד בסוף הספירה ופולסים רצופים. אופני הפעולה השונים של חמש היחידות (פורטים A, B ו-C ושני חלקי קוצב הזמן) נקבעים על-ידי תוכן האוגר COMMAND/ STATUS, לפי הציור הבא:



ציור 3-19 תיאור אוגר COMMAND ב-8155

האוגר COMMAND/STATUS בזמן קריאה ממנו מוסר את המידע הבא:

7	6	5	4	3	2	1	0
X	קוצב	INTE B	IBF B	INTR B	INTE A	BF A	INTR A

INTR (INTerrupt Request) - בקשת פסיקה.
BF (Buffer Full) - חוצץ מלא.
INTE (INTerrupt Enable) - איפשור פסיקה.

סיבית 6 מסמנת פסיקה של קוצב הזמן.

לפני השימוש ב-8155 יש לשלוח נתון לאוגר COMMAND/STATUS כדי לקבוע את אופני הפעולה של היחידות השונות.

קוצב הזמן הוא למעשה מונה בן 14 סיביות. טעינת המונה נעשית בשתי פניות לכתובות IO-3C ו-IO-3D שבדוגמא הקודמת. שתי הסיביות העליונות שב-IO-3D קובעות את MODE העבודה. לכניסת המונה מכניסים גל ריבועי כלשהו. יציאה המונה (TIMER OUT) היא בהתאם ל-MODE העבודה. הפעלת קוצב הזמן ועצירתו נעשית בפניה לאוגר הסטטוס (בדוגמא IO-38).

לקוצב ארבעה אופני עבודה:

- MODE0 - באופן עבודה זה יציאת המונה עולה ל-High למשך מחצית הספירה ויורדת ל-LOw למשך המחצית השניה. פעולה זו נעשית פעם אחת.
- MODE1 - אופן זה הוא כמו MODE 0 רק בסיום הספירה, נטען המספר שבקוצב שנית ומתחילה ספירה מחדש. MODE זה מוציא גל ריבועי רציף שיציאת המונה ב-Duty Cycle היא של 50%.

MODE2 - באופן עבודה זה, יציאת המונה נמצאת ב-High במשך כל הספירה עד הדופק האחרון. בדופק האחרון יורדת היציאה ל-Low למשך דופק אחד זה. פעולה זו נעשית פעם אחת.

MODE3 - אופן זה הוא כמו MODE 2 רק שהפעולה נעשית בצורה מחזורית.

3.7 חוצצים ו-BUS סטנדרטי

חיבור חוצצים ל-BUS:

כאשר מחברים מספר רב של רכיבי תמיכה למסלולי ה-CPU, יוצר הדבר העמסה על קוי ה-CPU. כדי להתגבר על בעיה זו, יש לחבר חוצצים בין יציאות ה-CPU למסלולים.

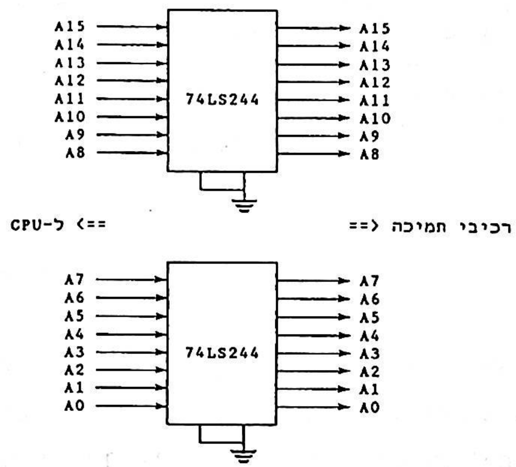
תפקידי החוצצים:

א. לאפשר צריכת זרם גבוהה יותר מאשר יכול לספק ה-CPU והגדלת ה-FAN OUT שלו.
ב. הפרדה בין ה-CPU לרכיבי התמיכה, כך שתקלה בחיבור הרכיבים לא תגרום נזק ל-CPU.
ג. עכבת המוצא הנמוכה של החוצצים מקטינה את הרעשים על הקוים השוכנים.

קיימים שני סוגי חוצצים - חד כיווניים לקוי הכתובת והבקרה, ודו-כיווניים לקוי הנתונים. לחוצצים יש בדרך כלל קו 'CS, המאפשר ניתוקם במידת הצורך. במקרה של שימוש בהם לצורך הגדלת ה-FAN OUT, אפשר לחבר קו זה באופן קבוע לאדמה.

חוצצים מקובלים הם ה-74LS244 וה-74LS365 כרכיב חד-כיווני לקוי הכתובת והבקרה, וה-74LS245 כרכיב דו-כיווני לקוי הנתונים.

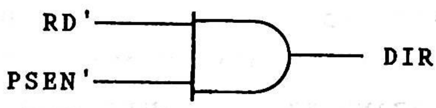
חיבור ה-74LS244 לקוי הכתובת יעשה בצורה הבאה:



ציור 3-20 חיבור חוצצים לקוי הכתובת

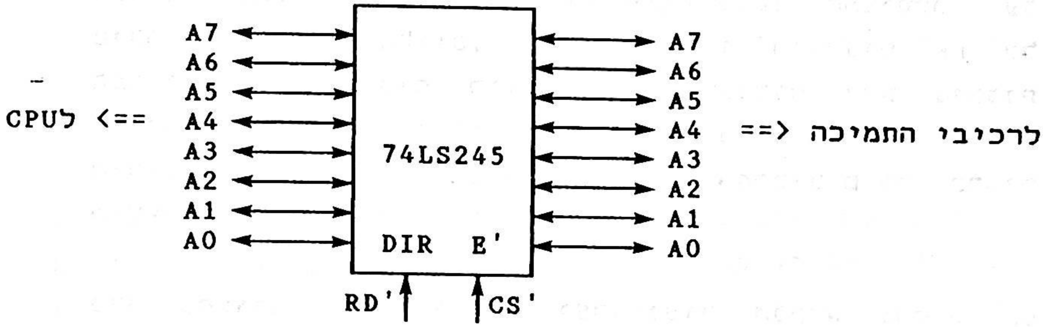
חיבור ה-74LS365 לקוי הבקרה נעשה בצורה דומה לחיבור ה-74LS244.

ה-74LS245 כולל כניסת כיוון (DIRection - DIR), שבעזרתה קובעים את כיוון זרימת האותות, מ-A ל-B או מ-B ל-A. ב-8085 מחברים לכניסת DIR את קוי 'RD. ב-8051 מחברים לכניסת DIR יציאת שער AND של 'RD ו-'PSEN.



ל-8088 יש קו מיוחד הנקרא 'DT/R שאותו מחברים לכניסת ה-DIR.

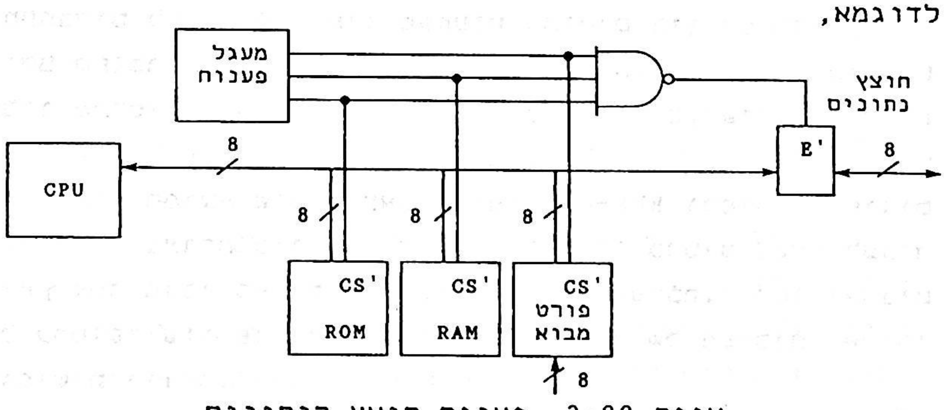
את כניסת ה-'E של הרכיב ניתן לחבר באופן קבוע לאדמה. החיבור מתואר בציור 3-21.



ציור 3-21 חיבור חוצצים לקוי הנתונים

חיבור קו 'E לאדמה אסור כאשר מחוברים רכיבים אל מסלול הנתונים של ה-CPU לפני החוצץ. כאשר יקרא ה-CPU מרכיבים אלה, הם יעבירו מידע אל מסלול הנתונים. גם החוצץ יעביר מידע, שימצא בכניסותיו, אל המסלול. לכן, במקרה כזה, מוסיפים למערכת מעגל פענוח, המעלה את קו 'E ל-High, כאשר מתבצעת פניה לאחד הרכיבים הפנימיים.

לדוגמא,



ציור 3-22 פענוח חוצץ הנתונים

ב-8088 יש להוסיף למעגל הפענוח קו Enable מיוחד היוצא מה-CPU והנקרא 'DEN.

סטנדרטי BUS

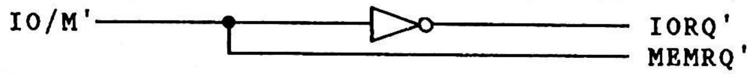
בעבר, כאשר תכננו מערכת מיקרו-מחשב המבוססת על מיקרופרוססורים מסוים, תכננו ובנו היצרנים את כל המרכיבים. המערכות היו מערכות סגורות ולא ניתנות להרחבה. לעתים קרובות החלקים ההיקפיים, כמו יחידות ה-I/O למיניהן, היו רבים יותר מאשר המרכיבים של כרטיס ה-CPU.

כדי להתגבר על בעיה זו יצרו מעגל הנקרא כרטיס אם (Mother board). כרטיס אם זה כולל מחברים (connectors) סטנדרטיים המחוברים במקביל. לאורך כרטיס האם עוברים קוים ומחברים במקביל את המחברים השונים. לדוגמא, פין 1 של מחבר 1 מחובר לפין 1 של מחבר 2 וכו'. המתכננים קבעו גם תפקיד לכל פין. חלק מיועד לקוי כתובת, חלק לקוי נתונים, קוי בקרה שונים, קוי אספקת מתח, קוי פסיקות וכו'. איפיון של כל פין ומספר הפינים, קובע את איפיון ה-BUS הסטנדרטי.

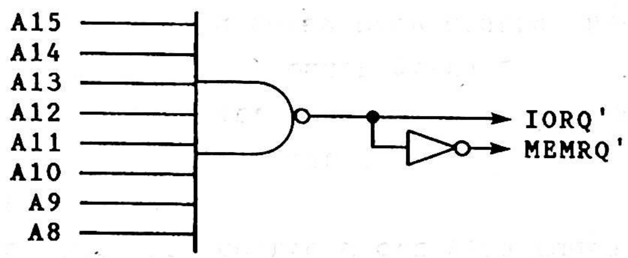
יחידות ה-I/O והרחבות הזכרון, נבנו בצורה של כרטיסים המתחברים למחברים. אין משמעות למיקום בין המחברים, כי כולם מחוברים במקביל. כל כרטיס מתייחס לאותות המגיעים אליו מכרטיס האם. המערכת כללה כרטיס CPU, שדאג לספק את האותות ולהפעיל את הכרטיסים. מתכנני הכרטיס דאגו להתאים את אותות המוצא שלו בהתאם לאיפיון ה-BUS ולמיקום בפינים השונים. בצורה כזו, ניתן היה להחליף כרטיס CPU ולעבור ל-µP אחר מבלי לשנות את שאר מרכיבי המערכת. קמו יצרנים של כרטיסי עזר שונים, שהתמחו בהרחבות של מערכות שהפכו לגמישות ביותר.

דוגמא, להתאמת אותות ראינו בערכות S-2085, S-2051 בצורת הפניה ליחידות I/O. ב-8085 יש קו הנקרא 'IO/M. בחברת SES הוחלט לעבוד עם קוים הנקראים 'IORQ (בקשת I/O)

ו-'MEMRQ (MEMory ReQuest). ב-8085 קו 'IORQ יורד ל-Low כאשר 'IO/M ב-High וקו 'MEMRQ יורד ל-Low כאשר קו 'IO/M ב-Low. כלומר, קו 'MEMRQ הוא למעשה קו 'IO/M. קו 'IORQ הוא ההיפוך שלו.



עבודה עם אותות כאלה מקובלת מאד ב-BUS סטנדרטי. לא עובדים עם קו בעל תפקוד כפול כמו 'IO/M, אלא מפרידים אותם לקוים שונים. ב-8051 אין קו המציין פניה ליחידות I/O כי הפניה היא במיפוי זכרון. לכן הוחלט על-ידי מתכנני S-2051 לקבוע את התחום FF00-FFFF כאיזור I/O. והם יצרו את המעגל הבא:



שתי הערכות כוללות מחברים זהים, בעלי תפקוד פינים אחיד, ולכן יחידות ההרחבה הן מודולאריות ויכולות להתחבר לכל אחת מהערכות מבלי שיצטרכו לבצע בהן שינוי כלשהו.

גם מחשבי ה-IBM PC כוללים BUS סטנדרטי בתוכם. מעגל ה-CPU כולל כרטיס אם ויחידות ההרחבה באות בצורה של כרטיסים. היום רוב יחידות ההרחבה נוצרות על-ידי יצרני משנה ולא על-ידי חברת IBM עצמה. הפניה ליחידות ה-I/O לדוגמא נעשית על-ידי שני קוים, 'IORD ו-'IOWR . קוים אלה כוללים בתוכם פענוח של תחום הכתובות 0300-03FF, הנחשב כתחום I/O למשתמש. כרטיס APPLIC-1 מפענח קוים אלה ויוצר מהם את קו 'IORQ.

הרעיון של BUS סטנדרטי קיים שנים רבות. ה-BUS הותיק ביותר היה BUS שנקרא S-100, שהיה מבוסס על כרטיסים גדולים מאד. עם התפתחות מעגלי LSI נוצר BUS שהוא פופולארי עד היום בתחום מערכות הבקרה, ונקרא STD-BUS. גם לחברות מוטורולה ואינטל יש BUS סטנדרטי משלהן. למוטורולה יש BUS הנקרא VME-BUS, ולאינטל יש BUS הנקרא MULTI-BUS.

3.8 שאלות ותרגילים

3.8.1 תכנן מעגל פענוח לפורט מוצא לכתובת IO-50. חבר למעגל פענוח זה את הרכיב 74LS374.

3.8.2 תכנן מעגל פענוח לפורט מבוא לכתובת IO-50. חבר למעגל פענוח זה את הרכיב 74LS244.

3.8.3 כתוב תוכנית המעבירה נתון מפורט מבוא לפורט מוצא בלולאה אינסופית. הפורטים הם אלה שתוארו בשאלות 3.8.1 ו-3.8.2.

3.8.4 תאר מעגל מלא לחיבור הרכיב 8255 בתחום הכתובות IO-14 עד IO-17. כתוב תוכנית הקובעת את פורט C כפורט מבוא ואת פורט A ופורט B כפורטי מוצא. התוכנית גם מעבירה נתון מפורט C לפורט A בלולאה אינסופית.

3.8.5 שרטט חיבור 8155 ל-CPU כך שתאי ה-RAM שלו יתפסו את הכתובות 8000-80FF.
איזה כתובות יתפסו פורטי הרכיב?

3.8.6 שרטט תכנון מערכת מיקרו-מחשב הכוללת:
CPU - מהסוג בו אתה משתמש.
Latch
EPROM 2764 לתחום הכתובות 0000-1FFF
6624RAM לתחום הכתובות 2000-3FFF
פורט מוצא בעזרת הרכיב 74LS273 בכתובת IO-1C

פורט מבוא בעזרת הרכיב 74LS244 בכתובת IO-1D
חוצצים ל-BUS, כאשר חוצץ הנתונים מתנתק כאשר יש
פניה לאחת היחידות המתוארות.

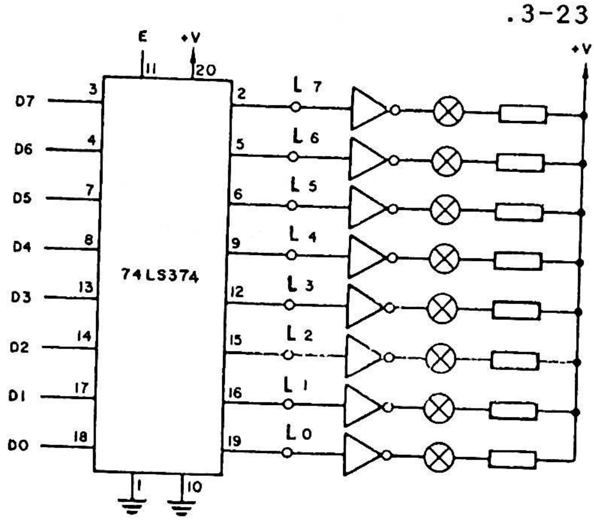
תרגיל

3.9 בניית פורט מוצא פשוט

מטרת התרגיל היא לבנות פורט מוצא פשוט בעזרת הרכיב 74LS374 ולחבר את יציאותיו ללדים של ערכת התרגול.

3.9.1 בנה את מעגל הפענוח המתואר בציור 3-3 על מטריצת החיבורים.

3.9.2 הוסף למעגל הפענוח את יחידת ה-74LS374 כמתואר בציור 3-23.



ציור 3-23 חיבור פורט מוצא ללדים של ערכת הניסויים

החיבור לדוחפים וללדים נעשה דרך נקודות החיבור במטריצה העליונה L0-L7.

3.9.3 כתוב תוכנית המעבירה לכתובת הפורט את המספר 0F. הרץ את התוכנית ובדוק אם ארבעת הנוריות המתאימות אכן נדלקות. אם לא נדלקו, דלג לסעיף 3.9.5 לשם איתור התקלה.

3.9.4 העבר מספרים שונים אל הפלט ובדוק אם ערכם הבינארי אכן מופיע בלדים.

3.9.5 הרץ את התוכנית בלולאה אינסופית ובדוק בעזרת סקופ, את האותות המופיעים ביציאות השערים. בדוק במיוחד את קו ה-CS.

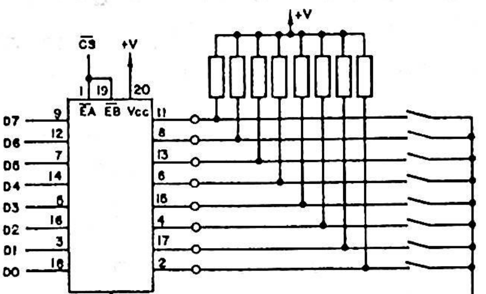
3.9.6 שרטט מעגל מלא הכולל CPU, Latch, מעגל פענוח ומעגל יחידת הפורט שבנית.

3.10 בניית פורט מבוא פשוט

מטרת התרגיל היא לבנות פורט מבוא פשוט בעזרת הרכיב 74LS244 ולחבר את כניסותיו למפסקי ערכת הניסויים.

3.10.1 בנה את מעגל הפענוח המתואר בציור 3-7 על מטריצת החיבורים. שים לב שאינך צריך לשנות הרבה במעגל הקודם, לכן בנה את פורט המבוא ליד פורט המוצא והיעזר במעגל הפענוח הקודם.

3.10.2 הוסף למעגל הפענוח את יחידת ה-74LS244 כמתואר בציור 3-24.



ציור 3-24 חיבור פורט מבוא למפסקי הערכה

החיבור למפסקי הערכה יתבצע דרך נקודות החיבור שלידם, S0-S7. נגדי משיכה כבר מחוברים לנקודות אלה.

3.10.3 כתוב תוכנית הטוענת את הצובר מפורט המבוא ולאחר מכן מוציאה תוכן הצובר לפורט המוצא. שים לב ששני הפורטים הם בעלי אותה הכתובת IO-FF.
קבע מספר בינארי כלשהו במפסקים, הרץ את התוכנית ובדוק אם המספר הופיע בנוריות הפלט. אם התוכנית אינה מתבצעת עבור לסעיף 3.10.4.

3.10.4 שנה את התוכנית לריצה בלולאה אינסופית. הרץ אותה. בדוק את האותות המופיעים ביציאות השערים. שינוי במפסקי הקלט צריך לגרום לאותו שינוי בנוריות הפלט.
בדוק בעזרת סקופ את האותות בכניסות וביציאות הרכיבים ובעיקר את קוי ה-'CS.

3.10.5 שרטט מעגל מלא הכולל את פורט המבוא, פורט המוצא, מעגל הפענוח, ה-CPU וה-Latch.

3.11 <u>שימוש בפורטי הערכה</u>

מטרת הניסוי היא לתרגל שימוש בפורט המוצא ובפורט המבוא של ערכת הניסויים על-ידי חיבורם אל הלדים של הערכה ואל מפסקי הערכה בהתאמה.

3.11.1 נתק את הרכיבים ממטריצת החיבורים.

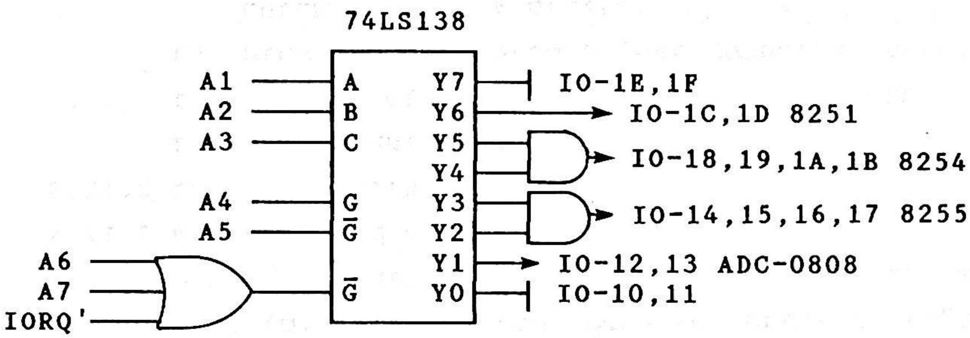
3.11.2 חבר את יציאות OUTPUT PORT B לכניסות הדוחפים ללדים L0-L7.

3.11.3 חבר את כניסות INPUT PORT A לחיבורי המפסקים S0-S7.

3.11.4 שנה את התוכנית מהניסוי הקודם כך שתעביר את תוכן המפסקים אל הנוריות בלולאה אינסופית דרך פורטי ערכת התרגול. כתובת הפורטים היא IO-11.

3.12 הפעלת פורט הניתן לתיכנות - 8255

כרטיס DSM-2200 כולל מספר רכיבים הניתנים לתכנות. חלקם תופסים 2 כתובות בתחום I/O וחלקם 4 כתובות בתחום זה. לכולם מעגל פענוח אחד, המבוסס על המפענח 74LS138 והמתואר במעגל הבא.



ציור 3-25 מעגל פענוח יחידות I/O של DSM-2200

הרכיבים ADC-0808, 8254, 8231 יוסברו בפרקים הבאים. לכניסות הכתובת CBA של המפענח מחוברים הקוים A3,A2,A1. לכן כל יציאה מופעלת בשתי כתובות I/O כאשר מופעלות כניסות האיפשור. ה-8255 צריך לתפוס 4 כתובות I/O ולכן קו 'CS שלו מתקבל מאיחוד, בעזרת שער AND, של שתי יציאות המפענח. תחום הכתובות שהוא תופס הוא IO-14H עד IO-17H. IO-14H הוא פורט A, IO-15H הוא פורט B, IO-16H הוא פורט C ו-IO-17H הוא אוגר הבקרה.

פורט B מחובר ל-DAC (Digital to Analog Converter), שנתאר בפרק 7. לשם תרגול עומדים לרשותנו פורט A ופורט C, שיציאותיהם/כניסותיהם מחוברים לסרגל מהדקים.

מעגל DSM-2150 הוא מעגל, הכולל 8 לדים ו-8 מפסקים זעירים. הוא ניתן לחיבור בצורה ישירה לסרגל המהדקים, כאשר המפסקים מחוברים לפורט C והנוריות לפורט A.

3.12.1 חבר את מעגל DSM-2150 לסרגל המהדקים של ה-8255 ב-DMS-2200.

3.12.2 רשום תוכנית המבצעת את השלבים הבאים:

א) קובעת את פורט A כפורט מוצא ואת פורט C כפורט מבוא בן 8 סיביות.

ב) קוראת את תוכן פורט C (מצב המפסקים) לצובר.

ג) מוציאה לפורט A (לנוריות) את תוכן הצובר.

ד) חוזרת לסעיף ב'.

3.12.3 הרץ את התוכנית.

3.12.4 שנה את המפסקים ובדוק אם הנוריות מגיבות.

3.12.5 שים לב, שהינך מקבל את התמונה המשלימה של המפסקים. "1" בכניסה מכבה את הנוריות ו-"0" מדליק אותה.

3.12.6 שנה את התוכנית כך ש-"1" ידליק את הנורית ו-"0" יכבה אותה.

3.13 <u>חיבור והפעלת 8155 (רשות)</u>

בסעיף 3.6 מתואר ה-8155 ואופן חיבורו. הוא כולל RAM ופורטים. בשאלה 3.8.5 נתבקשת לתכנן מעגל חיבור 8155 ל-CPU.

3.13.1 נתק את כרטיס DSM-2200 מהערכה S-2100.

3.13.2 בנה את מעגל ה-8155 על לוח התקיעה הראשי וחברו אל BUS הערכה בהתאם.

3.13.3 חבר את קוי פורט A למפסקים.

3.13.4 חבר את קוי פורט B לנוריות.

3.13.5 רשום תוכנית המבצעת:

א) קובעת את פורט A כפורט מבוא ואת פורט B כפורט מוצא.

ב) קוראת את פורט A לצובר.

ג) מוציאה את תוכן הצובר לפורט B.

ד) מדלגת לסעיף ב'.

3.13.6 הרץ את התוכנית, שנה את המפסקים ובדוק אם הנוריות מגיבות.

3.13.7 בפנל הערכה S-2100 נמצאות יציאות מתנדים. האחד בעל תדירות משתנה והיציאה מסומנת באותיות VAR (VARiable - משתנה). השני בעל תדירות קבועה של 50Hz ויציאתו מסומנת באותיות 50Hz.

3.13.8 חבר את יציאת ה-50Hz לכניסת המונה ולכניסת LED L0. חבר את יציאת המונה לכניסת L7 LED.

3.13.9 רשום תוכנית המוסרת פקודות היתחול ל-8155 כך שיציאתו תעלה ל-"1" כל 50 מניות ותרד ל-"0" כל 50 מניות נוספות בצורה מחזורית.
התוכנית מדלגת בסופה חזרה לתוכנית המוניטור.

3.13.10 הרץ את התוכנית.

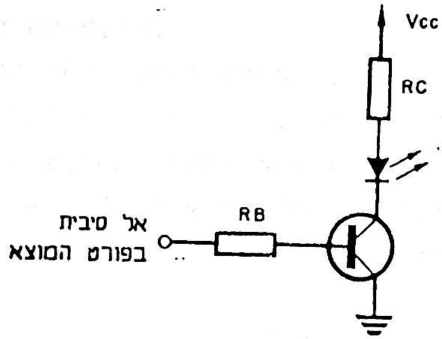
3.13.11 LED אחד (L0) צריך להבהב בקצב כזה, כך שלא ניתן יהיה להבחין בהיבהוב. הינך יכול לזהות את ההיבהוב בעזרת הבחון הלוגי ועוצמת האור הנמוכה. ה-LED השני (L7) צריך להבהב בתדירות של 0.5Hz. כלומר, לדלוק לשניה ולכבות לשניה.

תצוגה

4.1 LEDs ואופן חיבורם לפורט מוצא

ה-LED (Light Emitting Diode) הוא דיודה הפולטת אור כאשר זורם דרכה זרם. עוצמת האור תלויה בעוצמת הזרם דרכה. כדי לקבל עוצמת אור סבירה, מספיק להעביר דרכה זרם בין 5mA ל-10mA.

את ה-LED ניתן לחבר ישירות אל פורט המוצא, אבל רצוי יותר לחברו אל מעגל דוחף שיופעל על-ידי הפורט. מעגל הדוחף, הנקרא Driver, יכול להיות בנוי ממעגל טרנזיסטורי פשוט.



ציור 4-1 מעגל דוחף טרנזיסטורי ל-LED

RC ו-RB נקבעים בהתאם לזרם שרוצים שיעבור ב-LED (IC). נניח שרוצים לקבל IC=10mA. מפל המתח בהולכה על הדיודה ועל VCE (ברוויה) הוא 1.5V בערך. מכאן שעל RC יפול מתח של 3.5V.

$$RC = \frac{VRC}{IC} = 350\Omega$$

מקובל לחבר נגד של 330Ω כאשר רוצים לקבל זרם בסדר גודל של 10mA ו-680Ω כאשר רוצים לקבל זרם של 5mA.

הטרנזיסטור ממותג כאשר סיבית המוצא של הפורט, המחוברת אליו, נמצאת ב-High (VOH). VOH - מתח המוצא ה"גבוה" המקובל ברכיבים לוגיים הוא 3.3V בערך. IB הנדרש למיתוג הטרנזיסטור יחשב לפי β min.

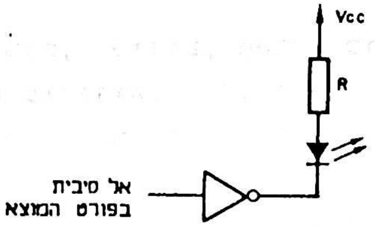
$$IB = \frac{IC}{\beta\ min}$$

ולכן

$$RB = \frac{VOH - VBE(SAT)}{IB}$$

RB מתקבל בסדר גודל של 5KΩ בערך, כאשר הקטנתו מעבר לכך גורמת להכנסת הטרנזיסטור לרוויה עמוקה. בתצוגה, בעיית זמני המיתוג לא קריטית בדרך כלל ולכן אין זה כורא.

ניתן גם למתג את ה-LED בעזרת מעגל דוחף משולב, כמו ה-7406 או ULN2001 בצורה הבאה:



ציור 4-2 מעגל דוחף משולב ל-LED

גם כאן ה-LED מאיר כאשר סיבית המוצא של הפורט נמצאת ב-High.

4.2 בבואה לפורט מוצא

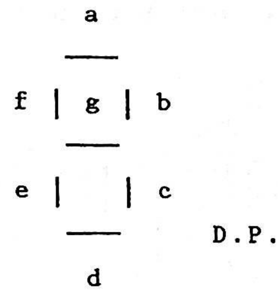
ברוב פורטי המוצא, אין אפשרות לקרוא את תוכנם. לדוגמא, נניח שרוצים לבצע "אור רץ" על שורת הלדים המחוברת לפורט מוצא. השיטה היא לבצע זאת בלולאה. מתהילים בהדלקת הנורית הימנית ומבצעים השהייה.

אם הפורט היה מתנהג כמו צובר, היינו פשוט מסובבים את תוכנו שמאלה וחוזרים לביצוע השהייה. אם הפורט היה מתנהג כמו תא בזכרון, היינו מעבירים את תוכנו לצובר, מסובבים שמאלה ומחזירים לתא.

מכיוון שאי אפשר לקרוא תוכן של פורט מוצא ולהעבירו לצובר, אין ברירה אלא לקבוע תא זכרון כבבואה של פורט המוצא. כל פעם שמעבירים נתון לפורט המוצא, מעבירים אותו נתון גם לתא זה. הקריאה של הנתון,לשם סיבובו, תיעשה מהתא.

4.3 תצוגת 7-SEG. והפעלתה ע"י ריבוב

תצוגת 7-SEG. מורכבת, כידוע, משבעה לדים בצורת ספרה 8, ולד שמיני כנקודה עשרונית.



ציור 3-4 ארגון ה-SEGments ביחידת .7-SEG

שמונת הלדים הם בעלי קתודה משותפת. את הקתודה המשותפת מחברים ל-GND. קיימות גם יחידות .7-SEG בעלות אנודה משותפת. את שמונת האנודות ניתן לחבר דרך מעגל דחיפה אל סיביות פורט מוצא. העלאת סיבית מוצא ל-High גורמת להדלקת אחד הלדים.

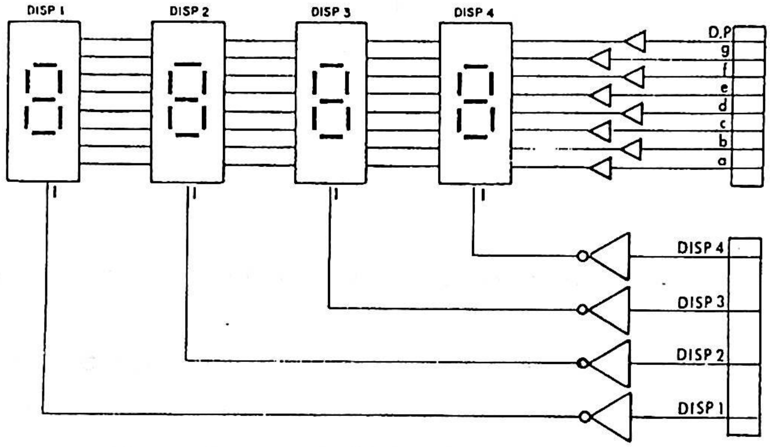
כאשר רוצים להפעיל תצוגה עם מספר יחידות .7-SEG, ניתן לחבר כל יחידה לפורט מוצא אחר. לשיטה זו שני חסרונות כבדים:

א. שימוש במספר רב של פורטי מוצא.
ב. צריכת זרם גדולה. בשיטה זו כל יחידת תצוגה מופעלת כל הזמן. הזרם הנצרך הוא סכום הזרמים שצורכות כל היחידות.

השיטה המקובלת להפעלת מספר יחידות תצוגה, היא הפעלתן בצורה דינמית.

בשיטה זו מחברים את שמונת הלדים של כל יחידות התצוגה במקביל, דרך מעגל דחיפה, אל יציאותיו של פורט מוצא אחד. נכנה פורט זה בשם פורט A. מספר בינארי המופיע ביציאותיו של פורט זה, מופיע למעשה בכניסות של כל יחידות ה-.7-SEG. את כניסות המתח של יחידות התצוגה מחברים, דרך

מעגל דחיפה, לפורט נוסף - פורט B (ראה ציור 4-4).



ציור 4-4 מעגל דחיפה ליחידות 7-SEG.

השלבים בהפעלת תצוגה בשיטה זו הם:

א. הוצאת המספר מיועד ליחידת התצוגה הראשונה בעזרת פורט A.
ב. סגירת מעגל ליחידה זו בעזרת פורט B.
ג. השהייה קצרה (ניתן לוותר עליה).
ד. ניתוק המעגל מיחידות התצוגה (00 לפורט B).
ה. הוצאת המספר המיועד ליחידת התצוגה השניה בעזרת פורט A.
ו. סגירת מעגל ליחידה זו בעזרת פורט B.
ז. השהייה קצרה.
ח. ניתוק המעגל מהיחידות וכן הלאה. לאחר הפעלת יחידת התצוגה האחרונה חוזרים ליחידת התצוגה הראשונה וחוזר חלילה.

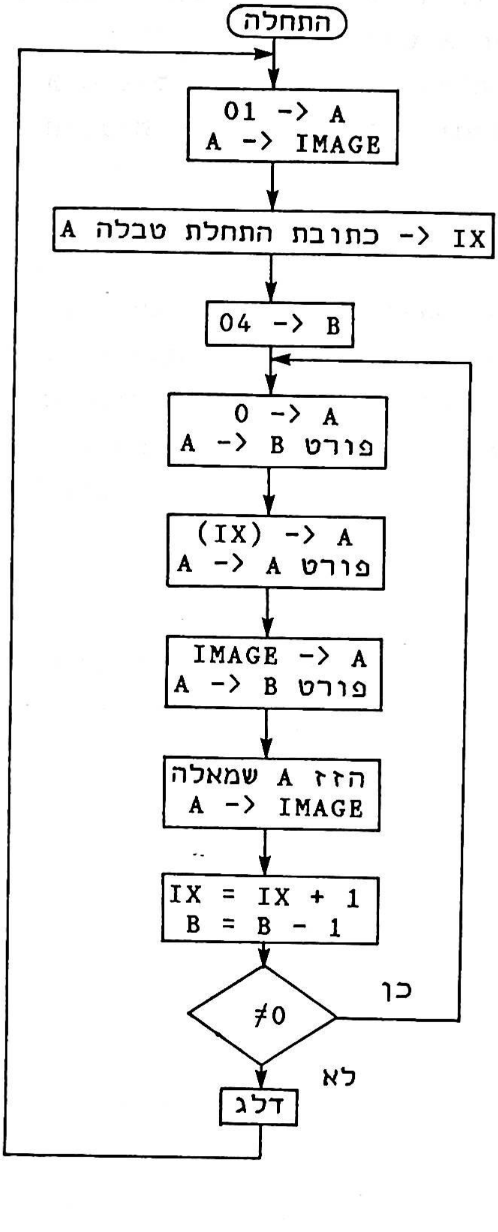
בשיטה זו, עסוק ה-CPU כל הזמן בהפעלת התצוגה. ניתן לבנות תוכנית זו בצורה של תוכנית משנה ולקרוא לה כל פעם

לשם הצגת המספרים פעם אחת ביחידת התצוגה.

לפניך תרשים זרימה לתוכנית, המציגה תוכן של ארבעה תאים בזכרון, בארבע יחידות תצוגה. פורט A מוצא את המספר לתצוגה. פורט B מפעיל את יחידת התצוגה הדרושה. אוגר B משמש כמונה. בתוכנית זו, המפעילה 4 יחידות תצוגה, הוא סופר עד 4.

התוכנית משתמשת בתאי זכרון מה-RAM. כותב התוכנית בוחר את התאים כרצונו. תא אחד ישמש כבבואה לפורט B ונכנה אותו בשם IMAGE. ארבעה תאים רצופים אחרים, ישמשו כטבלה לאיחסון המספרים המוצגים ביחידות התצוגה (טבלה A). IX משמש כ-Index Register (אוגר מציין), המצביע על התא שאת תוכנו עומדים להציג.

<u>תרשים זרימה א':</u>



במקרה של שימוש בתוכנית כתוכנית משנה, מוותרים על ה-דלג בסוף ורושמים במקומו "חזור מתוכנית משנה".

4.4 המרה מצופן BCD לצופן .7-SEG

המספרים הבינאריים הנמצאים בטבלה A ומיועדים לתצוגה, מפעילים את ה"לדים" שבתצוגה. כל סיבית מפעילה SEGMENT אחר. נקרא לצופן זה צופן .7-SEG. כאשר כותבים תוכנית המיועדת להוציא מספרים לתצוגה עשרונית, מוציאים מספרים בצופן BCD. נדרשת, אם כן, תוכנית שממירה טבלה של מספרים בצופן BCD (טבלה B), לטבלה של מספרים בצופן .7-SEG (טבלה A של התוכנית הקודמת).

כדי לבצע זאת נעזרים בטבלת חיפוש (טבלה זו נקראת Look (LUT - Up Table - טבלה C. טבלה זו מכילה את המספרים הבינאריים היוצרים את הספרות השונות ביחידות התצוגה. לדוגמא, בתא הראשון יהיה המספר הבינארי, היוצר את הספרה 0 בצופן .7-SEG, בתא השני את הספרה 1 וכן הלאה.

התוכנית פונה לתא הראשון בטבלה B (טבלת המספרים להמרה) ומעבירה את תוכנו לצובר. לאחר מכן, עוברת לתא שמספרו הסידורי בטבלה C שווה ל-A. לדוגמא, אם A שווה ל-3, התוכנית פונה לתא שמספרו הסידורי 3 בטבלה C (זהו התא הרביעי כי הראשון מספרו הסידורי הינו אפס). תא זה יכיל את המספר הבינארי היוצר את הספרה 3 בצופן .7-SEG. את תוכן תא זה מעבירה התוכנית לצובר ומשם לתא הראשון בטבלה A. לאחר מכן פונה התוכנית לתא השני בטבלה B. כך הלאה עד לסיום תאי טבלה B.

התוכנית בנויה כתוכנית משנה, מפני שקוראים לה פעם אחת, בכל פעם שנדרשת המרה.

כאמור, משתמשת התוכנית בשלוש טבלאות. כותב התוכנית חייב לקבוע את הכתובת ההתחלתית שלהן ואת אורכן. אורך טבלאות A ו-B הוא כמספר יחידות התצוגה.

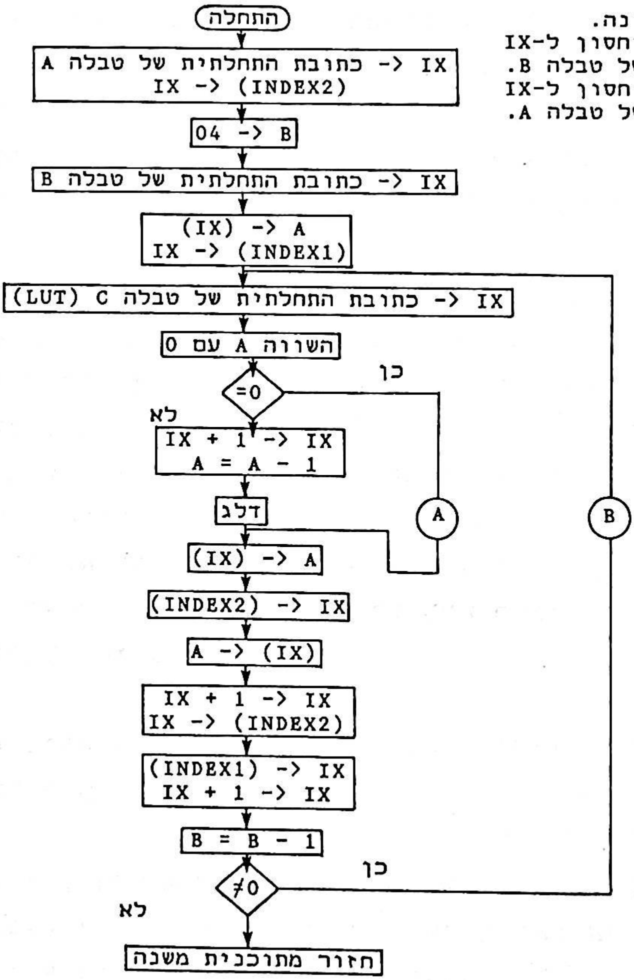
טבלה C (Look Up Table) מכילה עשרה מספרים. את טבלה C יש למלא יחד עם כתיבת התוכנית.

את טבלה B יש למלא לפני הקריאה לתוכנית. טבלה A היא הפלט של התוכנית.

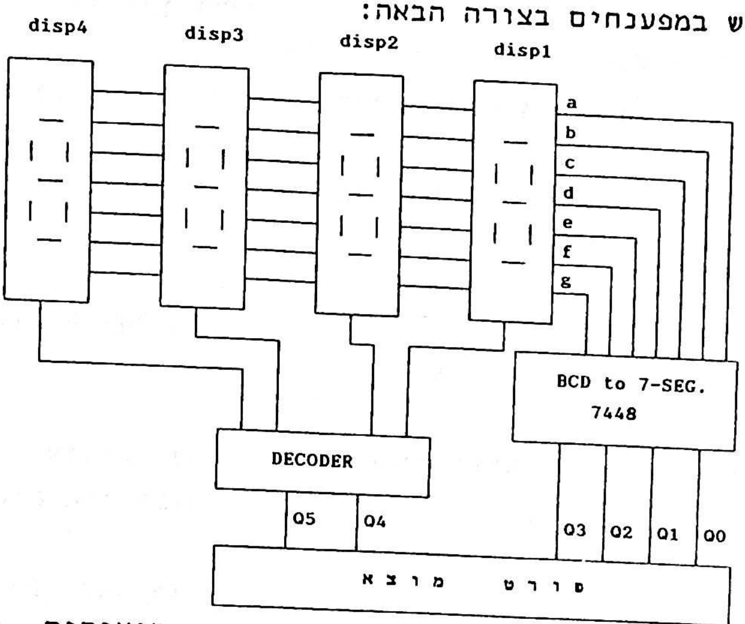
טבלה A צריכה להיקבע בתחום כתובות מתאים, כך שהתוכנית של תרשים זרימה א' תציג תוכן טבלה זו בתצוגה.

<u>תרשים זרימה ב':</u>

אוגר B - משמש כמונה.
INDEX1 - כתובת איחסון ל-IX המציין של טבלה B.
INDEX2 - כתובת איחסון ל-IX המציין של טבלה A.



ניתן לחסוך פורט אחד ואת השימוש בטבלת המרה על-ידי השימוש במפענחים בצורה הבאה:



ציור 5-4 תצוגה בריבוב בעזרת מפענחים

בשיטה זו מחברים את שבעת הלדים של כל יחידות התצוגה במקביל, דרך מפענח הנקרא BCD to 7-SEG., אל יציאותיו של פורט מוצא. מספר בינארי המופיע ביציאותיו של פורט זה, מופיע למעשה בכניסות של כל יחידות ה-7-SEG.. הקו המשותף של כל יחידת 7-SEG. מחובר ליציאת המפענח השני. BCD פירושו Binary Coded Decimal ומשמעותו מספר בינארי המציין מספר עשרוני.

המפענח 7448 מפענח מספר בינארי בן 4 סיביות ומפעיל את שבעת מקטעי יחידת התצוגה בהתאם לערך העשרוני. המפענח השני מפעיל, כל פעם יחידת תצוגה אחרת בהתאם למספר הבינארי בן שתי סיביות הניתן בכניסותיו.

השלבים בהפעלת תצוגה בשיטה זו הם:

א. הוצאת המספר המיועד ליחידת התצוגה הראשונה והפעלת יחידה זו. נדרש מספר בינארי בן 6 סיביות.

ב. הוצאת המספר המיועד ליחידת התצוגה השניה והפעלת יחידה זו וכן הלאה.

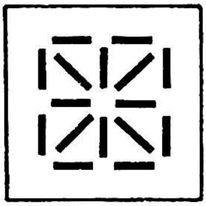
בשיטה זו, עסוק המחשב כל הזמן בהפעלת התצוגה. ניתן לבנות תוכנית זו בצורה של תוכנית משנה המציגה תוכן של ארבעה תאי זכרון או ארבעה משתנים במערך.

4.5 תצוגות אחרות

קיימות תצוגות נוספות כמו התצוגות האלפא נומריות. נזכיר חלק מהן בקצרה.

א. תצוגת .16-SEG:

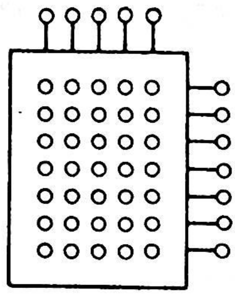
בנויה כמתואר בציור 4-6. מיתוגה דומה למיתוג יחידת .7-SEG.



ציור 4-6 תצוגת .16-SEG

ב. תצוגת מטריצת נקודות:

תצוגה זו בנויה במערך של נקודות. קיימים מספר סוגים של מערכים 5*7, 7*9, 9*11. לדוגמא, מערך 5*7.

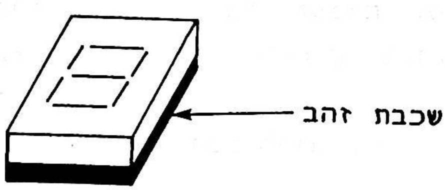


ציור 4-7 תצוגת מטריצת נקודות 5*7

מיתוג יחידה כזו מחייב שימוש במעגל מיתוג דינמי של כל עמודה בנפרד. מעגל כזה בנוי בחומרה ונעזר ב-ROM.

4.6 תצוגת גביש נוזלי LCD (Liquid Cristal Display)

תצוגת גביש נוזלי מבוססת על תמיסה מיוחדת הנתונה בין שני לוחות זכוכית. לוח זכוכית אחד מצופה בשכבת זהב דקיקה, והלוח השני בלוחות זהב דקיקים. שכבות הזהב דקות עד כדי שקיפות. לדוגמא, יחידת 7-SEG. מגביש נוזלי.



ציור 4-8 תצוגת 7-SEG. בגביש נוזלי

מוליכי זהב מקשרים את רגלי היחידה אל לוחות הזהב הדקים. התכונה המיוחדת של התמיסה היא, שכאשר מופיע שדה חשמלי דרכה, מסתדרות המולקולות שלה בצורה גבישית. זה גורם לתמיסה להפוך לשקופה. כאשר יסופק מתח לאחד הלוחות העליונים ביחס ללוח הזהב התחתון, תהפוך התמיסה שמתחתיו לשקופה. אם יונח רקע צבעוני כלשהו מתחתיו, הוא יראה. ברגע שיפסק המתח, תחזור התמיסה להיות עכורה והרקע יעלם. זו הסיבה, שתצוגה זו נקראת תצוגת גביש נוזלי.

ניתן למקם את לוחות הזהב בכל צורה שהיא, ולקבל תצוגות מגוונות ביותר. יתרונה הגדול של תצוגה זו, שאינה צורכת זרם. היא בנויה על מתח חשמלי ולא על זרם. חסרונה העיקרי של תצוגה זו, היא תלותה במקור אור חיצוני, וצריך לעמוד מולה על מנת לראות היטב. קיימות תצוגות LCD הכוללות מקור אור פנימי וצורת ארגון מיוחדת, המאפשרת הבחנה ברורה גם בזויות שונות ובאיכות טובה.

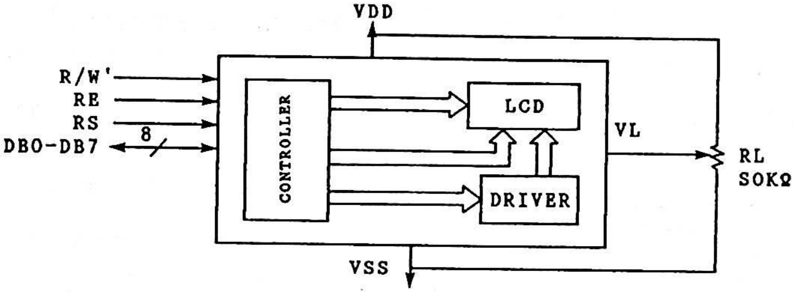
קיימות יחידות מיוחדות לתצוגות אלפא נומריות וגם תצוגות גרפיות. החל מתצוגה של שורה אחת בת שמונה תוים ועד תצוגה גרפית בגודל של מסך מחשב של 640*400 נקודות.

כדי לאפשר חיבור תצוגת ה-LCD בקלות למערכת מיקרו-מחשב, מספקים יצרני התצוגה יחד איתה מעגל בקר, המפעיל את לוחות התצוגה. מעגל הבקר מותאם להתחברות כרכיב תמיכה ל-CPU. הוא כולל גם RAM פנימי, שאליו כותבים את הנתונים. כל בית ב-RAM מייצג יחידת תצוגה אלפא נומרית אחת. היחידה כוללת גם מעגל פענוח מיוחד, הנקרא Character Generator (C.G). ה-C.G מתרגם את המספר הבינארי המופיע בבית, למספרים הבינאריים עבור העמודות השונות של יחידת התצוגה השייכת לבית זה.

נתאר חיבור יחידת תצוגת LCD מאד פופולארית, שיש לה יצרני משנה רבים, והמופיעה בתצורות רבות ומגוונות. אופן

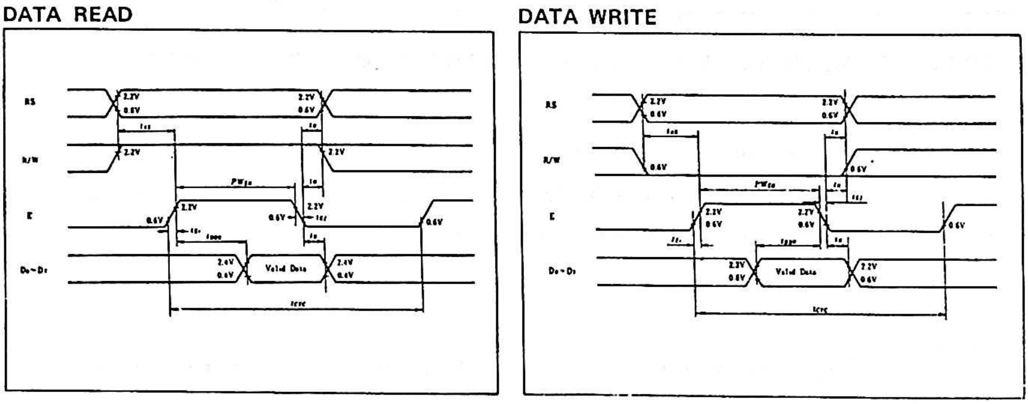
החיבור זהה, ללא תלות במספר יחידות התצוגה שהיא כוללת. היחידה תופסת שתי כתובות I/O. לכתובת הנמוכה כותבים פקודות בקרה שונות. לכתובת הגבוהה כותבים נתונים או קוראים נתונים ממנה. לדוגמא, נניח שרוצים לכתוב נתון בתא מסוים ב-RAM הפנימי. את כתובת התא מציינים בפקודה לכתובת הנמוכה ולאחר מכן כותבים את הנתון בכתובת הגבוהה.

דיאגרמת מלבנים של היחידה נראית בצורה הבאה:



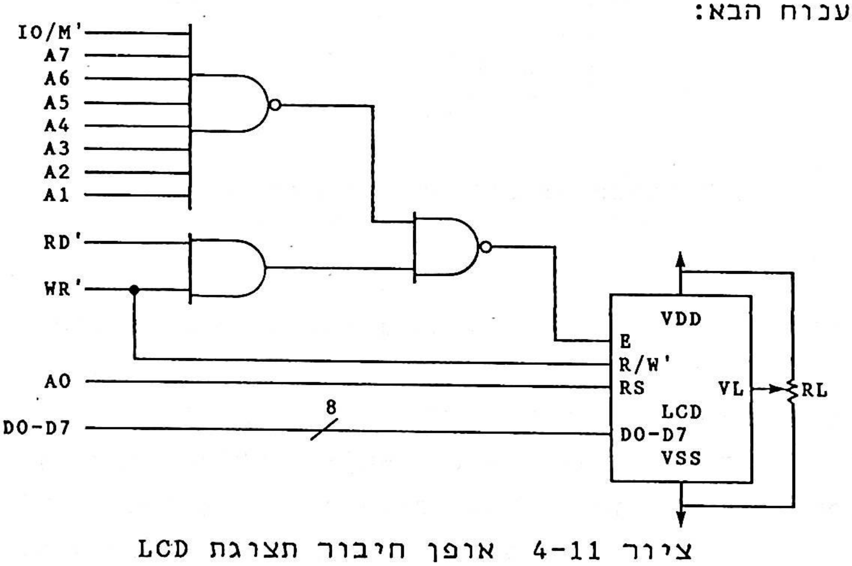
ציור 9-4 דיאגרמת מלבנים של תצוגת LCD

קוים DBO-DB7 מתחברים למסלול הנתונים DO-D7. כאמור, תופס הרכיב שתי כתובות בזכרון. RS מציין אם פונים לכתובת הגבוהה או הנמוכה. לכן, מחברים קו זה לקו AO. קו E הוא קו ה-CS למעשה בלוגיקה חיובית. הוא חייב להיות מתוזמן לקוי ה-'RD וה-'WR ומשלבים אותם במעגל הפענוח. קו 'R/W מציין אם כותבים ברכיב או קוראים ממנו. הדיאגרמה הבאה היא דיאגרמת הזמנים של הרכיב.



ציור 4-10 דיאגרמות זמנים של תצוגת LCD

נחבר יחידת תצוגה כזו לכתובות IO-FE,IO-FF, בעזרת מעגל הפענוח הבא:



ציור 4-11 אופן חיבור תצוגת LCD

קיימות שיטות נוספות להפעלת תצוגה. קיימות שיטות המשלבות תוכנה עם חומרה או פועלות בחומרה בלבד. אחת השיטות הנפוצות להפעלת תצוגה בחומרה משתמשת ברכיב 8279. מכיוון שרכיב זה מטפל גם בלוח מקשים נתאר את השימוש בו בפרק הבא.

4.7 שאלות ותרגילים

4.7.1 רשום תרשים זרימה ותוכנית ליצירת "אור רץ" בפורט IO-12H.

47..2 המר את תרשים זרימה א' לתוכנית כאשר פורט A הוא בכתובת IO-11H ופורט B הוא בכתובת IO-12H. את כתובת טבלת המספרים המוצגים קבע כרצונך בהתאם ל-RAM במערכת המיקרו-מחשב.

4.7.3 המר את תרשים זרימה ב' לתוכנית כאשר את כתובת טבלת המספרים להמרה קבע כרצונך בהתאם ל-RAM במערכת המיקרו-מחשב שברשותך. את המספרים המומרים העבר לטבלה שהשתמשת בה בתוכנית שבשאלה (4.7.2). קבע גם את כתובת טבלת החיפוש.

4.7.4 בכניסות יחידות ה-7-SEG., מחובר מפענח BCD to 7-SEG., כמתואר בציור 4-5 . פורט המוצא המפעיל מעגל זה הוא IO-11. רשום תוכנית המציגה: תוכן של ארבעה תאים ב-RAM בארבע יחידות התצוגה. את תוכן התא הראשון מציגה ביחידה הימנית ביותר, את תוכן התא השני מציגה ביחידה הבאה, וכך הלאה כאשר בין הצגה להצגה מבצעת השהייה באורך של מחצית השניה בערך.

4.7.5 שרטט מעגל מיתוג של ארבע יחידות תצוגה מסוג מטריצת נקודות 5*7.

4.7.6 שרטט חיבור תצוגה LCD לתחום הכתובות IO-58H,59H.

תרגול

4.8 8212 כפורט מוצא פשוט

מטרת הניסוי להפעיל את שמונת הלדים בעזרת 8212, המחובר כפורט מוצא. לשם שימוש ב-8212 כפורט מוצא פשוט, אין צורך לעשות דבר, מלבד להתחבר ליציאותיו. קוים 'CLR ו-'STB מחוברים עם נגדי משיכה ל-+V. הפורט תופס את הכתובת IO-12H.

4.8.1 חבר את יציאות OUTPUT PORT D לכניסות הדוחפים של הלדים.

4.8.2 שרטט מעגל חשמלי הכולל את פורט המוצא, הדוחפים, הלדים והנגדים המחוברים אליהם.

4.8.3 הוצא מידע כלשהו לפורט המוצא בעזרת תוכנית קצרה ובדוק שהלדים נדלקים.

4.8.4 חבר לזמן קצר, בעזרת חוט, את קו 'CLR ל-GND. בדוק שהלדים כבו.

4.8.5 רשום תוכנית ליצירת "אור רץ" בנורידות הפלט והרץ אותה.

4.9 הפעלת תצוגת .7-SEG

4.9.1 תקע בלוח התקיעה הראשי את הכרטיס ,DSM-2130 הכולל 4 יחידות תצוגת .7-SEG ולוח מקשים.

4.9.2 חבר את הכניסות Q0-Q5 ליציאות של פורט מוצא .PB0-PB5 חבר VCC ו-GND למעגל.

סיביות Q0-Q3 מחוברות לכניסות מפענח BCD to .7-SEG. סיביות Q4,Q5 מחוברות למפענח, הקובע איזה יחידה מופעלת, כמתואר בציור 4-5.

4.9.3 חבר את כניסות פורט מבוא A למפסקים S0-S5 בהתאמה.

4.9.4 רשום תוכנית הקוראת את המפסקים מפורט A לצובר ומוציאה את תוכן הצובר לפורט מוצא B בלולאה אינסופית. הרץ את התוכנית.

4.9.5 שנה תחילה את ארבעת המפסקים הימניים ביותר (LSB). את ארבעת המפסקים השמאליים השאר במצב ."0"

הספרה הימנית ביותר מבין ארבעת יחידות התצוגה צריכה להשתנות בהתאם. התבונן מה קורה כאשר הנך קובע מספרים גדולים מ-9.

4.9.6 שנה עתה את המספקים .S4,S5 התצוגה תעבור מיחידת תצוגה אחת לאחרת.

4.9.7 רשום תוכנית המבצעת את השלבים הבאים:

א) טוענת לצובר את תוכן פורט מבוא A (מצב המפסקים).

ב) מאפסת את ארבע הסיביות השמאליות. מבצעת AND בין הצובר עם .0F

ג) מוציאה את המספר לפורט מוצא .B

ד) מוסיפה ל-A .10H

ה) מוציאה את המספר לפורט מוצא .B

ו) מוסיפה ל-A .10H

ז) מוציאה את המספר לפורט מוצא .B

ח) מוסיפה ל-A 10H.

ט) מוציאה את המספר לפורט מוצא B.

י) מדלגת להתחלה (לסעיף א').

4.9.8 הרץ את התוכנית.

המספר המופיע במפסקים צריך להופיע בכל יחידות התצוגה. שינוי המפסקים משנה את תוכן כל היחידות. למרות שבכל פעם מופעלת יחידת תצוגה אחרת, הן נראות כדולקות כל הזמן.

4.9.9 נסה לבצע שלבים ג' עד ט' בעזרת לולאת מנייה.

4.9.10 רשום את התוכנית המופיעה בתרגיל 4.9.4, המעבירה לארבע יחידות התצוגה תוכן של ארבעה תאים מהזכרון. לכל יחידה תוכן של תא אחר. טען בתאים מספרים שונים והרץ את התוכנית. בדוק שאכן מופיעים מספרים אלה בארבע יחידות התצוגה.

4.10 הפעלת יחידות התצוגה בצורה ישירה (רשות)

לעתים רוצים ליצור בעזרת יחידות ה-7-SEG. צורות שונות מהספרות אותן יוצר המפענח BCD to 7-SEG. במקרה כזה צריכים להפעיל כל מקטע ב-7-SEG. ישירות מפורט המוצא.

במעגל DSM-2130 ניתן לבצע זאת. מצד ימין של המעגל יש 8 נקודות חיבור, המסומנות באותיות a-g,dp וכן ארבע נקודות dsp1-dsp4.

4.10.1 הלחם חוטים חד גידיים לנקודות חיבור אלה.

4.10.2 הוצא את המפענחים מהתושבות.

4.10.3 חבר את הנקודות a-g,dp ליציאות פורט מוצא B.

4.10.4 חבר את הנקודות dsp1-dsp4 ליציאות פורט מוצא D.

4.10.5 חבר את מפסקי הערכה S0-S7 לפורט מבוא A.

4.10.6 רשום תוכנית המבצעת:

א) מוציאה לפורט מוצא D את המספר 01.

ב) קוראת לצובר את תוכן פורט מבוא A.

ג) מוציאה את תוכן הצובר לפורט מוצא B.

ד) חוזרת לסעיף ב'.

4.10.7 הרץ את התוכנית

4.10.8 שנה את המפסקים ועקוב אחר התצוגה.

הפעל בתחילה כל מקטע בנפרד ובדוק איזו סיבית מפעילה כל מקטע. רשום זאת במחברתך.

צור צורות שונות בעזרת תצוגת שבעת המקטעים.

4.10.9 רשום והרץ את התוכנית המשלבת את תרשימי זרימה א' ו-ב' המתוארים בסעיפים 4.3 ו-4.4.

מפסקים ומקשים

5.1 קריאת מצב מפסקים וזיהוי המפסק המופעל

בפרק 3 תרגלנו חיבור מפסקים אל פורט מבוא, וכיצד לקרוא את מצב המפסקים ולהעביר לפורט מוצא, המוציא אותו לנוריות. כדי להגיב להפעלת מפסק כלשהו, יש לזהות עליה ל-"1" בסיבית המקבילה לו, במספר הבינארי שהתקבל מהמפסקים. הבדיקה נעשית בעזרת הפעולה הלוגית AND.

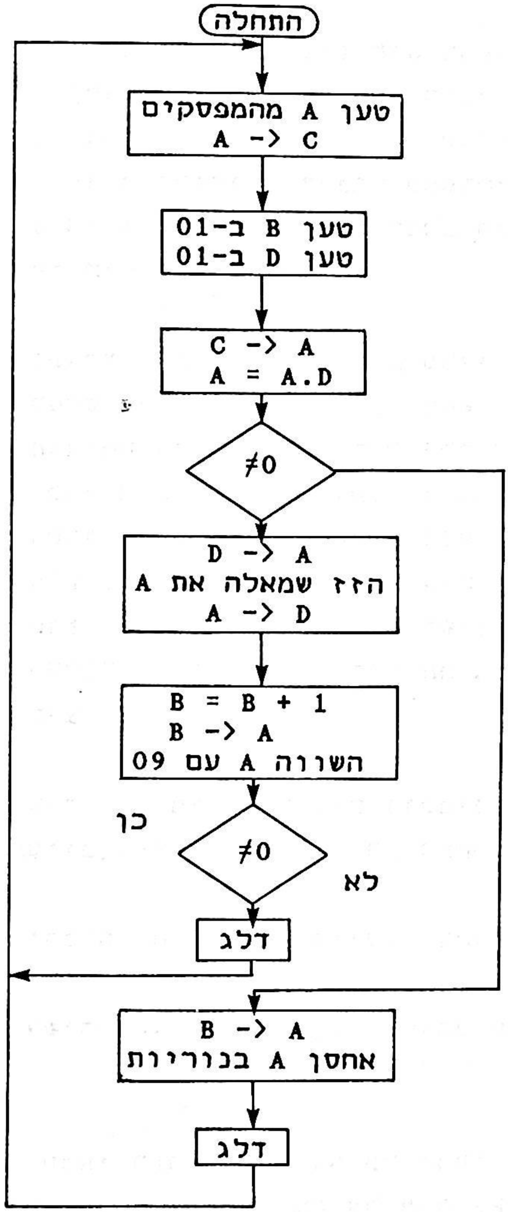
לדוגמא, נניח שרוצים לבדוק סיבית D1 (הסיבית השניה) והמספר נמצא בצובר. מבצעים פעולת AND על הצובר, במיעון מידי, עם המספר הבינארי 00000010. האפסים יאלצו את הסיביות המקבילות להם בצובר לרדת ל-"0", לא משנה מה היה מצבו הקודם. רק סיבית D1 תישאר במצבה הקודם.

אם לאחר הפעולה מכיל הצובר אפס (דגל האפס עלה ל-"1") אזי D1 היה "0". אם לאחר הפעולה בצובר יש מספר השונה מאפס (דגל האפס נמצא ב-"0") אז D1 היה "1". בדיקת מצב הסיבית תהיה בהתאם לדגל האפס.

נכתוב תוכנית המוציאה לנוריות הפלט את מספר המפסק הנלחץ. לדוגמא, לחיצה על המפסק הרביעי תגרום שיופיע בנוריות הפלט המספר 04H (ולא המספר 08H שהיה מופיע שם, אילו העברנו את מצב המפסקים ישירות אל הפלט).

הפקודה AND משנה את תוכן הצובר, לכן יש לשמור את תוכן הצובר באוגר ולאחר כל בדיקה לטעון את הצובר מחדש.

תרשים זרימה א':



שים לב שהתוכנית בודקת שמונה מפסקים, בזה אחר זה בלולאה אינסופית. חשוב מה קורה כאשר שני מפסקים מופעלים.

5.2 סריקת מפסקים רגעיים

מפסק רגעי (Momentary Switch) הוא מפסק שיש לו מצב נורמלי מסוים (סגור או פתוח) ומצב רגעי - המשלים של המצב הנורמלי. בלחיצה על המפסק הוא עובר למצבו הרגעי ונשאר במצב זה עד לשחרורו. דוגמא למפסקים רגעיים הם מקשי מחשב. לצורך נוחיות, נניח שהמצב הנורמלי הוא מצב LOW והמצב הרגעי הוא מצב HIGH.

כדי להגיב לפעולת מקש, יש לבדוק עליית מקש ל-HIGH ולפעול בהתאם למקש הנלחץ. לחיצה על מקש במחשב, גורמת להדפסת אות מתאימה במסך. לחיצה כפולה על אותו מקש, גורמת להדפסה כפולה של אותה האות, בהמשך אותה שורה. פעולת המחשב, בתגובה לעליית המקש ל-HIGH , מסתיימת לפני ירידת המקש ל-LOW (המחשב מהיר מאצבע אדם והמקש המיכני). במקרה כזה, חוזר המחשב לסרוק את המקשים ומוצא את המקש הלחוץ עדיין ב-HIGH ושוב מגיב. כתוצאה מכך, נקבל שורה של תוים עקב לחיצה אחת.

פתרון פשוט לבעיה (קיימים פתרונות נוספים) הוא לרשום את עליית המקש המסוים ל-HIGH ולהגיב רק לאחר ירידתו ל-LOW.

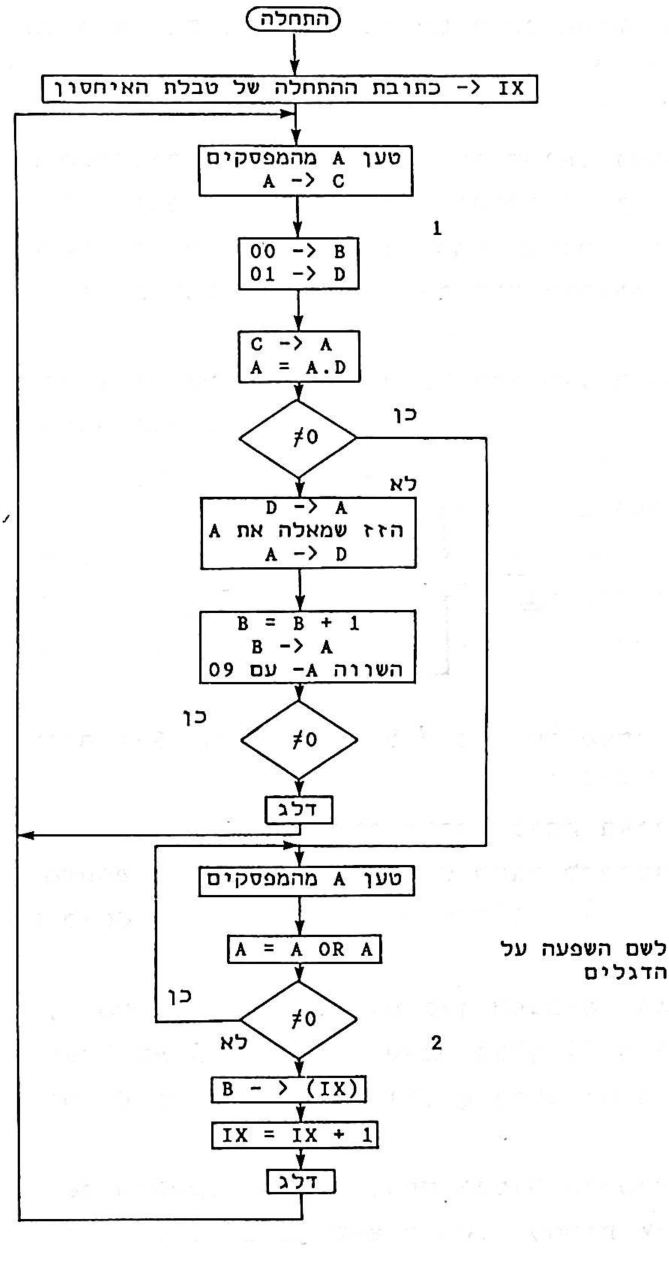
נשכלל את התוכנית מהסעיף הקודם. במקום להוציא את המספרים הסידוריים של המפסקים הנלחצים אל הנוריות, נאחסן אותם בשורת תאים בזכרון. התוכנית רצה בלולאה אינסופית.

ניתן להוסיף תנאי כזה, שלחיצה על מפסק מסוים תגרום להפסקת התוכנית. במחשב, לחיצה על מקש <CR> (Carriage Return) גורמת למעבר לשורה חדשה ולפענוח השורה הרשומה.

לסימנים 1 ו-2 שבתרשים זרימה ב' כתייחס בסעיף הבא.

<u>תרשים זרימה ב':</u>

IX - אוגר מציין

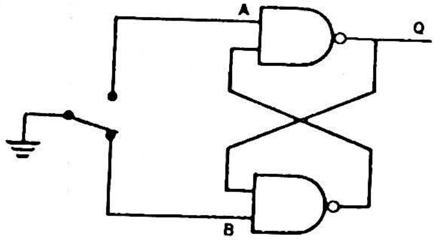


5.3 ביטול ריטוט מקשים (Debouncing)

בכל מפסק קיימת בעיית ריטוט, המופיע בזמן מעבר של המפסק. לחיצה על המפסק, גורמת למגעיו לרטוט (להתחבר ולהתנתק) מספר פעמים עד להתייצבותם במצב החדש. גם שחרור המפסק גורם לתופעת הריטוטים.

אם נריץ את התוכנית שבסעיף הקודם כמו שהיא, נקבל שלחיצה אחת על מקש גורמת להדפסת מספרו הסידורי בטבלה מספר פעמים. המחשב מהיר יותר מזמן ההתייצבות של המפסק. קיימות שתי שיטות לפתרון הבעיה בחומרה ובתוכנה.

השיטה לפתרון בעיה בחומרה, היא על-ידי חיבור Flip-Flop לכל מקש, בצורה הבאה:



ציור 5-1 מעגל לביטול ריטוט של מפסק

אם השערים הם מסוג TTL, הרי שהדק כניסה באויר מהווה "1". אם משתמשים בלוגיקת CMOS יש לחבר לכניסות A ו-B נגדי משיכה ל-VCC.

בזמן מעבר, מתקבלים הריטוטים בין המגעים (A או B) למירווח, אבל לא בין המגעים עצמם (בין A ל-B). במצב המתואר בציור, B ב-"0" ו-A ב-"1", Q במצב זה ב-"0".

בזמן מעבר של המפסק, הדק B נפתח ונסגר חליפות. הדק A נשאר כל הזמן ב-"1", בזמן ש-B רוטט. למרות ש-B מחליף

מצב מספר פעמים מ-"0" ל-"1" וחזרה ל-"0", Q כשאר ללא שינוי (בדוק זאת). רק כאשר A מתקצר לאדמה בפעם הראשונה, Q מחליף מצב ל-"1". ריטוטים נוספים ב-A שוב לא יגרמו לשינוי ב-Q והוא ישאר ב-"1" עד לקיצור B לאדמה שנית. כל זאת בזכות המשוב הקיים בין השערים.

לפתרון זה של הבעיה, יש מחיר גבוה. הפתרון מחייב שימוש במפסקים מחליפים וחיבור מעגל אלקטרוני לכל מקש.

לפתרון הבעיה בתוכנה, קיימות שתי שיטות. שיטה אחת מבוססת על לקיחת שורת דגימות מהמקשים. רק כאשר מבחינים שסידרה רצופה של דגימות נותנת אותו המצב מתייחסים לשינוי מצב זה. בשיטה זו, ההתייחסות היא רק למצבים היציבים של המפסקים ולא למצבי הריטוט שלהם.

השיטה השניה פשוטה יותר, אבל גם היא אמינה למדי. שיטה זו מבוססת על הוספת לולאה השהייה לאחר כל הבחנה בשינוי מצב למשך זמן ארוך יותר מזמן הריטוטים. שיטה זו מחייבת הכרת משך זמן הריטוטים של המפסק. אין טעם לתת זמן ארוך מדי, כי אז לא יובחנו לחיצות מהירות על המקשים.

את לולאות ההשהייה יש להוסיף לתרשים זרימה ב' במקומות המסומנים ב-1 וב-2.

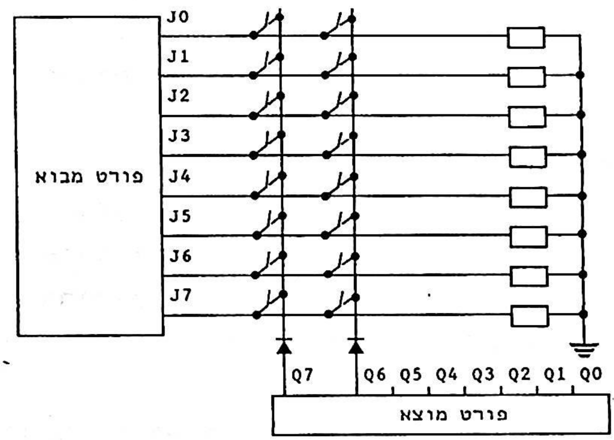
5.4 לוח מקשים (Keyboard)

בשיטות שתוארו לסריקת המקשים, חיברנו כל מפסק אל סיבית בפורט. הדבר מחייב שימוש בפורטי מבוא כמספר המקשים חלקי שמונה.

לוח מקשים מורכב ממספר רב של מפסקים ושיטה זו של סריקה בזבזנית למדי בחומרה. למשל, 64 מקשים (לוח מקשים מלא) מחייב שימוש בשמונה פורטים, ומספר חוטים רב.

כדי להתגבר על בעיה זו, עורכים את לוח המקשים במערך (בצורה של מטריצה) ומשתמשים בפורט מוצא ובפורט מבוא. לדוגמא נתאר מערך של 16 מקשים.

להפעלת מערך זה, זקוקים ל-2 סיביות של פורט מוצא ו-8 סיביות של פורט מבוא. ניתן גם להשתמש ב-4 סיביות פורט מוצא ו-4 סיביות פורט מבוא. כשתמש בצורת הארגון הראשונה.



ציור 2-5 ארגון 16 מקשים במערך

<u>עקרון סריקת המקשים מבוסס על האלגוריתם הבא:</u>

א. הוצא "1" לקו Q7 של פורט המוצא.
ב. סרוק את קוי הכניסה של פורט המבוא לפי התוכנית של תרשים זרימה ב'. אם כל המקשים בשורה זו אינם לחוצים, כל הכניסות תהיינה ב-"0".
ג. אם אחד המקשים בשורה זו לחוץ יתקבל "1" בקו שלו, עבור לסעיף ו'.
ד. הוצא "1 " לקו Q5 של פורט המוצא.

ה. חזור על סעיפים ב' ו-ג'.

ו. מספר המקש הנלחץ הוא מספר הקו בפורט המוצא המחובר למקש הנלחץ כפול 8, פלוס מספר הקו בפורט המבוא שם נמצא "1".

בשרטוט הנראה בציור 5-2 קל לראות את אופן הרחבת לוח המקשים לעד 64 מקשים, על-ידי הוספת שורות מקשים נוספות.

ניתן לחסוך בסיביות פורט מוצא על-ידי שימוש במפענח (Decoder). לדוגמא, כדי לקבל 8 שורות מפסקים, מחברים 3 סיביות מפורט המוצא למפענח 1 מ-8, ואת יציאות המפענח נחבר לשורות.

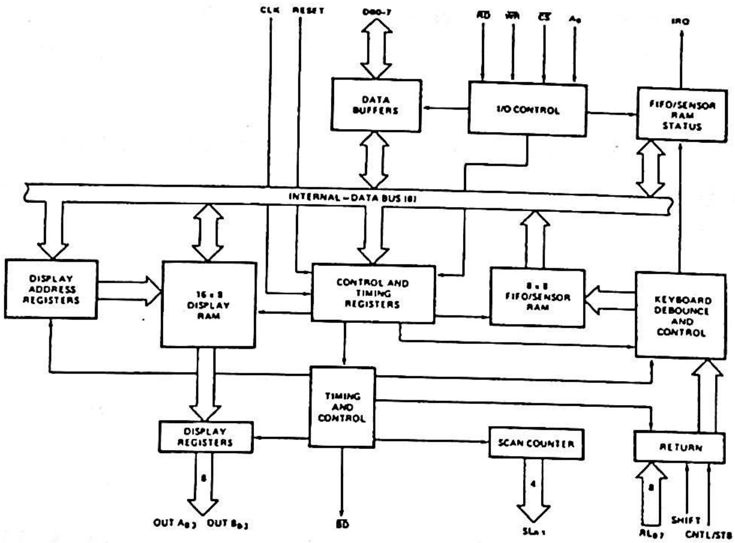
5.5 8279 כממשק תצוגה ולוח מקשים

ה-8279 הוא רכיב המתחבר אל BUS ה-CPU ומאפשר הצגת תוים בתצוגת ה-.7-SEG וסריקת המקשים בו זמנית. הוא מבצע את הצגת התוים וסריקת המקשים בחומרה מבלי להיעזר ב-CPU לשם כך. בצורה זו הוא משחרר את ה-CPU לריצה על תוכניות אחרות.

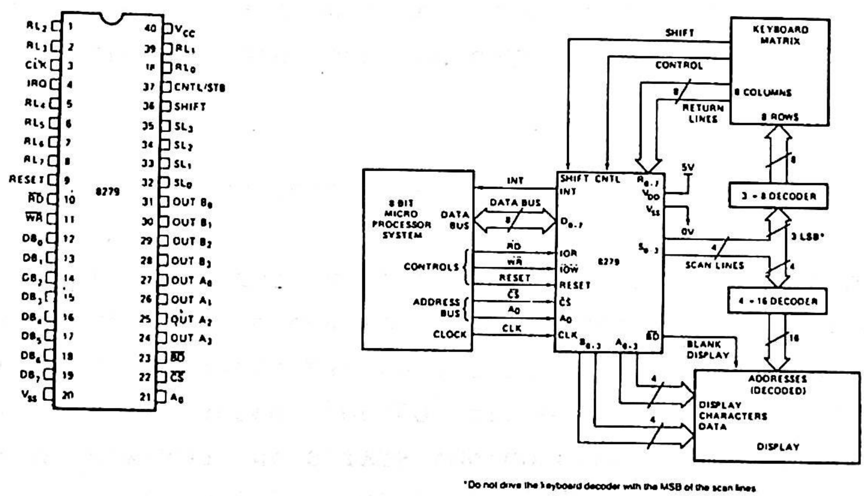
השימוש ברכיב מפשט במידה ניכרת את תוכנת המערכת.

הרכיב תופס שתי כתובות במפת הזכרון או במפת I/O בהתאם למעגל הפענוח. מעגל הפענוח_מפענח את קוי הכתובת מלבד את קו A0. קו A0 מתחבר ישירות לרכיב. בצורה זו מתקבל 'CS בפניה לשתי כתובות. הפורט הגבוה (כאשר A0=1) משמש להעברת COMMAND או STATUS והפורט הנמוך להעברת DATA בהתאם.

דיאגרמת המלבנים של הרכיב וצורת השימוש בו מתוארים בציורים הבאים:



ציור 5-3 דיאגרמת מלבנים של ה-8279



ציור 5-4 אופן חיבור ה-8279 ללוח מקשים ולתצוגה

ל-8279 מצבי הפעלה שונים ומגוונים. נתאר שלושה מהם. השאר ניתן למצוא בדפי הנתונים של הרכיב.

א. היתחול ה-8279:

בפניה לאוגר ה-COMMAND, שלושת הסיביות הגבוהות של המלה קובעים את סוג הפניה ומטרתה. חמשת הסיביות הבאות משמשות כפרמטרים. בסך הכל יש שמונה פניות לאוגר ה-COMMAND עבור פונקציות שונות.

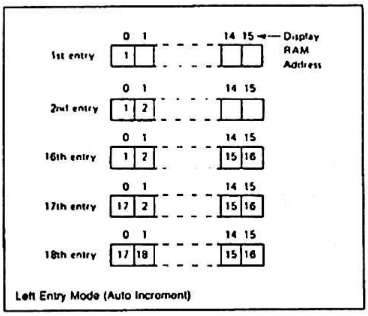
המלה לאוגר						תפקיד
000	D	D	K	K	K	קביעת MODE SET
001	P	P	P	P	P	קביעת קצב סריקת המקשים והתצוגה ביחס ל-CLK הנכנס.
010	A1	X	A	A	A	קריאה מה-FIFO RAM
011	A1	A	A	A	A	קריאה מה-Display RAM
100	A1	A	A	A	A	כתיבה ב-Dispaly RAM
101	X	DA	WB	BA	BB	פניה לחצי מקוי התצוגה A או B
110	CD	CD	CD	CF	CA	איפוס (clear) של קוי התצוגה
111	E	X	X	X	X	קביעת Error Mode Set

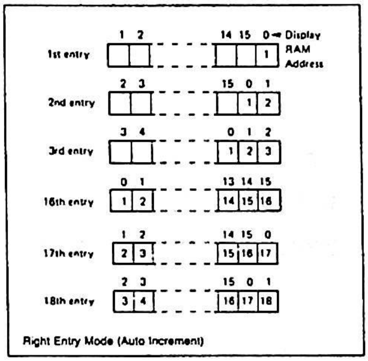
כאשר מהתחלים את הרכיב, יש לקבוע MODE SET. קביעת SET MODE נעשית על-ידי כתיבת המלה 000DDKKK. הסיבית DD מציינות את MODE הפעלת התצוגה.

DD
00 - 8 8 bit character display - left entry
01 - 16 8 bit character display - left entry
10 - 8 8 bit character display - right entry
11 - 16 8 bit character display - right entry

המשמעות של right entry או left entry היא שבמצב של הכנסת נתונים אוטומאטית ל-RAM התצוגה (ראה בסעיף הבא), הנתונים יכנסו מימין לשמאל או משמאל לימין, כמתואר

בצירורים הבאים:





ציור 5-5 אופני הכנסת הנתונים ל-RAM התצוגה

הסיביות KKK מציינות את MODE הפעלת לוח המקשים באפשרויות הבאות:

KKK

000 - Encoded scan keyboard - 2 key lockout

001 - Decoded scan keyboard - 2 key lockout

010 - Encoded scan keyboard - N key roll over

011 - Decoded scan keyboard - N key roll over

100 - Encoded scan sensor matrix

101 - Decoded scan sensor matrix

110 - Strobed input, encoded display scan

111 - Strobed input, decoded display scan

במושג Encoded מתכוונים שקוי העמודות של מערך המקשים מגיעים ממפענח שכניסותיו הן כקוים SL0-2 ומקבלים מערך של 8*8 * קו shift וקו CNTRL. במושג Decoded מתכוונים שהקים SL0-3 משמשים כקוי העמודות ולכן מקבלים מערך של 8*4 בלבד.

במושג 2 Key lockout מתכוונים, שהרכיב בודק האם 2 מקשים כלחצו בו זמנית ומתעלם מהם. הרכיב מתגבר בעצמו על בעיית

ריטוט המקשים, על-ידי כך שהוא סורק את לוח המקשים פעמיים נוספות לאחר זיהוי מקש שנלחץ. רק אם לא חל שינוי במצב המקש, הוא מכניס את מספרו ל-FIFO RAM. אם במצב בדיקת הריטוט נלחץ מקש נוסף, הוא מתעלם משני המקשים.

במושג N key roll over מתכוונים שקיימת אפשרות ללחיצה על מספר מקשים וכל לחיצה על מקש תיכנס ל-FIFO RAM.

במושג Sensor matrix מתכוונים שמחברים למערך רכיבים שאין להם בעיית ריטוט, כמו יציאות שערים או פורטים. הרכיב סורק אותם ללא ניסיון ביטול הריטוט.

במושג Strobed input, הכוונה שהרכיב סורק את מערך המקשים, רק לאחר עליית קו CNTRL, המשמש כקו Strobe, ל-High.

ברירת המחדל של הרכיב, כלומר המספר שנמצא בו במצב היתחול ולאחר RST הוא המספר 00001000. בדוק את משמעותו.

פעולה נוספת, שצריך לבצע בהיתחול הרכיב זה קביעת תדירות סריקת התצוגה והמקשים. זה קובע גם את ההשהייה לביטול ריטוט המקשים. ל-8279 מספקים לכניסת ה-CLK גל ריבועי. זה יכול להיות ממקור גל ריבועי כלשהו או מיציאת השעון של ה-CPU. תדירות מומלצת לעבודת הרכיב היא 100KHz. ניתן להגיע לתדירות זו גם ממקורות בעלי תדירות גבוהה יותר.

הרכיב כולל מחלק, הניתן לתכנות על-ידי מלת ה-COMMAND 001PPPP. לדוגמא, אם מספקים לו גל בתדירות של 2MHz, כותבים בו את מלת ה-COMMAND 001 10100, הגורמת לחלוקה ב-20 ויוצרת גל של 100KHz.

פעולה אחרת, שצריך לבצע בצד ה-CPU הפעם, היא לאפשר פסיקה. כלומר, עם הופעת בקשת פסיקה על-ידי הרכיב, שתגיע לאחר לחיצה על מקש, צריך ה-CPU לפנות לתוכנית פסיקה שתקרא נתון זה מה-FIFO RAM. המלה FIFO משמעותה First In First Out. כלומר, ראשון נכנס ראשון יוצא. ראה סעיף ג'.

ב. פניה לתצוגה:

ל-8279 RAM פנימי בן 16*8 BIT המשמש את התצוגה. הוא מציג את תוכן תאי ה-RAM בצורה דינמית, בזה אחר זה בתצוגת ה-.7-SEG. יחידות ה-.7-SEG מחוברות כמתואר בפרק הקודם. בכל יחידת תצוגה יש 8 לדים (7-SEGments ונקודה עשרונית). 8 הלדים מחוברים במקביל. כניסות ה-COMMON של יחידות התצוגה מחוברות ליציאות מפענח 1 מ-16 של קוי ה-SL (Scan Lines). הרכיב, בעזרת קוי ה-SL, פונה כל פעם ליחידת תצוגה אחרת ומעביר אליה מה-RAM את תוכן התא המתאים.

כל סיבית בנתון המועבר, מדליקה או מכבה (1-מכבה, 0-מדליק) את הלד המתאים. לדוגמא, הנתון FB (11111011) מדליק את הקו האמצעי בלבד.

ניתן לחבר אל הרכיב תצוגה שמספר יחידותיה קטן מ-16, על-ידי שימוש בחלק מיציאות המפענח. אי אפשר להשתמש ברכיב לתצוגה בעלת יותר מ-16 יחידות.

כדי להציג נתון מסוים באחת מיחידות ה-.7-SEG, יש לפנות לפורט הגבוה ולהעביר אליו את הנתון A A A A A1 0 0 1. AAAA מציין את כתובת התא אליו פונים. לאחר מכן פונים לפורט הנמוך ומוסרים את הנתון שרוצים שיכנס לתא הנ"ל. ה-8279 יציג כל הזמן את תוכן התאים בתצוגה.

כאשר מעבירים שורה של נתונים לתאי ה-RAM, בזה אחר זה, אין צורך לפנות לפורט הגבוה בכל פעם (לשם העברת כתובת תא חדש). מספיק שבפעם הראשונה נכניס בסיבית A1 - "1" לוגי. לאחר כל פניה לפורט הנמוך, תגדל הכתובת AAAA אוטומאטית ב-1. אם A1=0, אזי הכתובת אינה משתנה אלא לאחר פניה לפורט הגבוה עם כתובת אחרת.

לדוגמא, תוכנית המדליקה את הקו האמצעי ב-4 מיחידות התצוגה. נניח, שמעגל הפענוח קובע את כתובת הפורט הגבוה ל-IO-05 וכתובת הפורט הנמוך ל-IO-04, כמו ב-S-2085 וב-S-2051.

8051	8085	
START: MOV A,#90H	START: MVI A,90H	טען A ב-10010000
MOV DPTR,#FF05H		
MOVX @DPTR,A	OUT 05H	העבר A לאוגר הבקרה
MOV A,#FBH	MVI A,FBH	טען A ב-11111011
MOV DPTR,#FF04H		
MOV R7,#06	MVI B,06H	טען מונה (B) ב-06
LOOP1: MOVX @DPTR,A	LOOP1: OUT 04H	הוצא תוכן A לפורט נמוך
DJNZ R7,LOOP1	DCR B	חסר 1 מהמונה
	JNZ LOOP1	אם שונה מאפס, דלג ל-LOOP1
LOOP2: SJMP LOOP2	LOOP2: JMP LOOP2	לולאה אינסופית כדי לראות את התצוגה

התוכנית מסתיימת בלולאה אינסופית, על מנת לא לשנות את התצוגה.

כפי שרואים בציור 4-5, קוי המידע היוצאים לתצוגה מורכבים משתי רביעיות A0-3, B0-3. שתי הרביעיות מרכיבות מלה בת 8 סיביות, המגיעה מ-RAM התצוגה. הסיבה שזה מצוין בצורה כזו היא כי ניתן להפעיל כל רביעיה בנפרד. ניתן לחבר כל רביעיה למפענח BCD to 7-SEG. ולקבל עד שתי

שורות בנות 16 יחידות תצוגה כל אחת. ניתן לפנות ולשנות תוכן כל רביעיה ב-RAM התצוגה מבלי לפגוע בתוכן הרביעיה השניה. ה-8279 כולל אופן פניה מיוחד ל-RAM התצוגה לשם כך.

ג. פניה למקשים:

כמתואר בציור 4-5, יש ל-8279 קו בקשת פסיקה. קו זה מחברים לאחד מקוי הפסיקה של ה-CPU בהתאם לעדיפות שרוצים להעניק לו. את המקשים מחברים במטריצה אל קוי ה-RL (Return Lines) מצד אחד ואל יציאות מפענח 1 מ-8 מצד שני. המפענח את שלושת קוי ה-LSB של קוי ה-Scan Lines של ה-8279 (ראה ציור). סך הכל, ניתן לחבר מטריצה של 8*8 מקשים.
ה-8279 סורק את המקשים כל הזמן, בלי תלות בעבודת ה-CPU. לחיצה על אחד המקשים גורמת ליצירת בקשת פסיקה על-ידי הרכיב, והכנת מספר המקש הנלחץ בפורט הנמוך.

תוכנית הפסיקה של ה-CPU צריכה לפנות ל-FIFO RAM ולקחת משם את מספר המקש.

קריאה מה-FIFO RAM נעשית על-ידי מלת ה-COMMAND 010A1XAAA. X פירושו סיבית חסרת חשיבות (don't care). הסיביות AAA מציינות את מספר התא ב-FIFO RAM אליו רוצים לפנות. לאחר מלת COMMAND זו פונים לפורט הנמוך וקוראים משם את המידע, שיגיע מהתא המצוין. אם A1=0 זה מחייב, שבכל פניה נוספת ל-FIFO RAM יש לציין את כתובת התא המבוקש. אם A1=1 ניתן לקרוא את התאים ברציפות. פעם אחת כותבים את כתובת התא הראשון שרוצים לקרוא ממנו. לאחר מכן, כל קריאה מהפורט הנמוך תביא את הנתון המבוקש ותגרום למצביע פנימי להצביע על התא הבא בתור.

ב-S-2085 מחובר ה-8279 לקו הפסיקה RST5.5 ותופס את

הכתובת 04,05. תוכנית הפסיקה (הניתנת למיסוך) קוראת את מספר המקש ומאחסנת אותו בתא 38FEH. כדי לבדוק אם נלחץ מקש חדש, מכניסים לתא 38FE את המספר FF בזמן ההיתחול. לאחר מכן דוגמים את התא. אם מוצאים שם מספר שונה מ-FF, סימן שנלחץ מקש. אוספים את המספר ומכניסים שוב לתא את המספר FF.

בנוסף להכנסת המספר FF לתא 38FE יש לאפשר את בקשת הפסיקה על-ידי הפקודות:

```
MVI A,08
SIM
EI
```

ב-S-2051 מחובר ה-8279 לקו פסיקה INTO ותופס את הכתובות FF04,FF05. תוכנית הפסיקה מאחסנת את מספר המקש בתא 7F שב-RAM הישיר. גם כאן יש לאחסן FF בתא זה לפני השימוש בלוח המקשים. לאחר מכן יש לפנות לתא 7F ולבדוק אם המספר הוא FF. אם הוא שונה מ-FF, אזי נלחץ מקש וזהו מספרו. לאחר איסוף הנתון מהתא, יש לטעון את התא שנית ב-FF.

לפני השימוש בסריקת המקשים יש לאפשר את בקשת הפסיקה בעזרת הפקודות:

```
MOV IE,#83H
MOV IP,#01
```

הסיבה לטעינת IE ב-83 ולא ב-81 היא כדי לאפשר גם פסיקת TIMERO לצורך ריצה בצעד יחיד (ראה ספר משתמש S-2051).

5.6 שאלות ותרגילים

5.6.1 המר את תרשים זרימה א' לתוכנית, כאשר המפסקים מחוברים לפורט מבוא בכתובת IO-11H.

5.6.2 המר את תרשים זרימה ב' לתוכנית. המפסקים מחוברים כמו בשאלה 5.6.1.

5.6.3 הוסף לתוכנית שבסעיף 2 לולאות השהייה במקומות המסומנים ב-1 וב-2 בתרשים הזרימה. רשום את התוכנית מחדש.

5.6.4 רשום תוכנית לקריאת לוח מקשים בהתאם לאלגוריתם המתואר בסעיף 5.4.

5.6.5 שרטט מעגל חשמלי מפורט בהתאם למעגל המתואר בציור 5-4, לחיבור 8279 לתחום הכתובות IO-04,05. ה-8279 מפעיל תצוגה של 4 יחידות 7-SEG. ולוח מקשים של 2*8.

5.6.6 רשום תוכנית המאחסנת בארבע יחידות התצוגה תוכן של ארבעה מתאי הזכרון. התוכנית מבצעת זאת פעם אחת ומדלגת לתוכנית המוניטור.

5.6.7 קרא בספר המשתמש של הערכה שברשותך, וראה כיצד מתבצעת קריאה מהמקשים והפעלת התצוגה.
רשום תוכנית הממתינה ללחיצה על מקש ומציגה אותו ביחידת תצוגה מס' 3. התוכנית רצה בלולאה אינסופית.

תרגיל

5.7 זיהוי מפסק נלחץ

מטרת התרגיל היא לקרוא את מצב מפסקי הערכה ולזהות את המפסק הנלחץ, על-ידי העברת מספרו הסידורי לנוריות הפלט. כל זאת בעזרת התוכנית לתרשים זרימה א'.

5.7.1 חבר את מפסקי הערכה לפורא מבוא A.

5.7.2 חבר את יציאות פורא מוצא B לכניסות הלדים של הערכה.

5.7.3 כתוב תוכנית לתרשים זרימה א'.

5.7.4 הרץ את התוכנית בלולאה אינסופית. שנה את מצב המפסקים ובדוק אם מספר המפסק המורם אכן מופיע בנוריות הפלט.

5.7.5 בדוק ורשום מה קורה כאשר שני מפסקים מורמים.

5.8 סריקת מפסקים רגעיים

מטרת הניסוי היא לקרוא את מצב מפסקי הערכה ולזהות את המפסק הנלחץ כמו בניסוי הקודם, כאשר במקום להעביר את מספרי המפסקים הנלחצים לנוריות, להעבירם לטבלה בזה אחר זה, בהתאם לתרשים זרימה ב', ולזהות את בעיית הריטוט (Bouncing).

5.8.1 חבר לפורט מבוא A ולפורט מוצא B את המפסקים והלדים בהתאמה.

5.8.2 כתוב תוכנית לתרשים זרימה ב', כאשר הנך קובע כתובת התחלתית לטבלה שלשם תעביר התוכנית את

מספרי המקשים הנלחצים.

5.8.3 הרץ את התוכנית בלולאה אינסופית.

5.8.4 הרם בכל פעם מפסק אחר והורד אותו חזרה. רשום לעצמך את סדר הרמת המפסקים.

5.8.5 הפסק את התוכנית. פנה לטבלה ובדוק אם המספרים הרשומים שם מתאימים לסדר הלחיצה על המפסקים.

5.8.6 יש סיכוי סביר, שתקבל על כל לחיצת מפסק, את מספרו הסידורי מספר פעמים. הסבר מדוע.

5.9 ביטול ריטוט מפסקים בתוכנה

מטרת התוכנית היא למצוא את אורך לולאות ההשהייה הדרושות כדי לבטל את תגובת המחשב לריטוט המפסקים, והוספת לולאות השהייה בהתאם לתוכנית מתרגיל 5.8.

5.9.1 שפר את התוכנית מתרגיל 5.8 על מנת לבטל את התופעה שתואר בסעיף 5.3. הוסף לולאת השהייה. התחל בלולאות השהייה קצרות מאד.

5.9.2 הרץ את התוכנית ובדוק אם המחשב מגיב עדיין לריטוט המפסקים.

5.9.3 אם כן, הגדל את ההשהייה וחזור על סעיף 2, עד לביטול תגובת המחשב לריטוט המפסקים.

5.9.4 חשב את גודל ההשהייה הנדרשת ב-μS ורשום אותה.

5.10 חיבור לוח מקשים במטריצה

מטרת התרגיל היא לחבר מטריצה של 16 מקשים הנמצאת בכרטיס DMS-2130 אל פורטי המערכת וזיהוי המקשים הנלחצים.

5.10.1 חבר את יציאות פורט מוצא B אל הלדים בערכה.
5.10.2 תקע את מעגל DMS-2130 בלוח התקיעה הראשי.
5.10.3 חבר את המטריצה של 16 המקשים המאורגנים בצורה של 2*8 אל פורט מוצא D ואל פורט מבוא A בהתאם לציור 2-5.
5.10.4 רשום תוכנית בהתאם לאלגוריתם שבסעיף 5.4, שתוציא את מספר המקש הנלחץ אל נוריות הפלט.
5.10.5 הרץ את התוכנית.
5.10.6 כדי להפעיל מקש יש ליצור קצר בין מגעיו.
5.10.7 קצר אחד המקשים ובדוק אם מספרו הסידורי מופיע בנוריות הפלט.
5.10.8 הפעל מספר מקשים ובדוק את התוצאה בפלט.

5.11 הפעלת תצוגה ומקשים בעזרת 8279

ניסוי זה מבוסס על הערכות S-2051 או S-2085, שבהן התצוגה והמקשים מחוברים אל ה-CPU בעזרת הרכיב 8279, כמתואר בסעיף 5.5. בשתי הערכות מחובר ה-8279 לקו פסיקה. עם לחיצה על מקש, מאחסנת תוכנית הפסיקה את מספרו בתא בזכרון (תא 38FE ב-S-2085 ותא 7F ב-S-2051). נכנה תא זה בשם KBRD.

5.11.1 רשום תוכנית המבצעת:
א) היתחול ואיפשור בקשת הפסיקה.
ב) דגימת תא KBRD בלולאה והשוואת תוכנו עם FF.

ג) אם תוכן התא שונה מ-FF הוצאת המספר לכוריות פורט 00, הכנסת FF לתא KBRD וחזרה לסעיף ב'.

5.11.2 הרץ את התוכנית.

5.11.3 לחץ על המקשים ורשום לך איזה מספר מתקבל מכל מקש.

5.11.4 רשום תוכנית המבצעת:

א) היתחול ה-8279 להפעלת התצוגה.

ב) קריאת מצב המפסקים מפורט 00.

ג) הוצאת המספר ליחידת התצוגה הימנית ביותר.

ד) חזרה לסעיף ב'.

5.11.5 הרץ את התוכנית.

5.11.6 שנה את מצב המפסקים, ורשום לך איזה מקטע מפעילה כל סיבית.

5.11.7 שפר את התוכנית שבסעיף 5.11.4 כך שתיעזר בטבלת חיפוש ותמיר מספרים מ-0 עד F לספרות ואותיות ביחידת התצוגה.

5.11.8 הרץ את התוכנית.

5.11.9 שנה את ארבעת המפסקים הימניים (ארבעת השמאליים קבע ב-"0") ובדוק את התוצאה בתצוגה.

5.11.10 שלב את שתי התוכניות (מקשים ותצוגה) לתוכנית אחת, הקולטת מספר מלוח המקשים ומוציאה אותו לתצוגה. הרץ ובדוק.

5.11.11 שפר את התוכנית כך, שהכנסת מספר לתצוגה תבצע הזזת הספרות שמאלה אוטומאטית (Right Entry). הרץ ובדוק.

פרק 6

מונה/קוצב זמן

6.1 מונה/קוצב זמן (Timer/counter) - 8253/8254

קוצב זמן הוא מונה, הסופר דפקים המגיעים אליו בקצב קבוע, כך שלמעשה הוא מודד זמן. ההבחנה בשמות נעשית בהתאם לתפקוד של הרכיב. אם מחברים לכניסת המונה אות המגיע בקצב אחיד, מתפקד הרכיב כקוצב זמן (Timer). אם מחברים את כניסת המונה למקור חיצוני של אותות, שמייצר דפקים לא בהכרח בקצב אחיד, יתפקד הרכיב כמונה אירועים (Counter).

בפרק זה נתאר את הרכיבים 8253/8254. שני הרכיבים זהים לחלוטים כמעט. כל ההבדל ביניהם הוא שה-8254 מסוגל לעבוד בתדרים גבוהים יותר מה-8253. בכל הפרק נשתמש בשם 8254 אבל, הכוונה גם ל-8253. אלה רכיבים פופולאריים למדי בתחום זה.

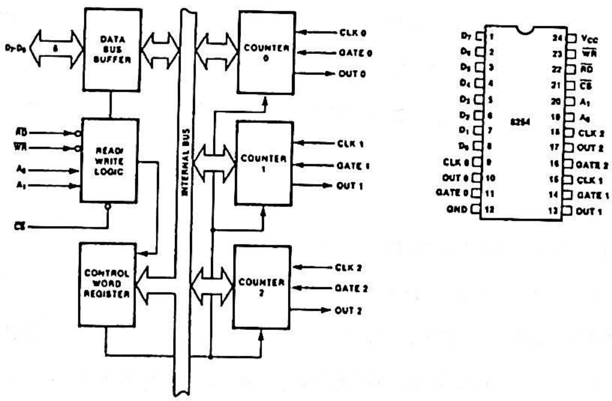
ל-8254 שלושה מונים/קוצבי זמן, המהווים כלי עזר חשובים ביותר. כלים אלה חוסכים בתכנות של לולאות השהייה מדויקות, של דגימת אירועים באמצעות פניה לפורט ובדיקה ומשחררים את ה-CPU לביצוע משימות אחרות.

נמחיש את חשיבות קוצב הזמן בדוגמא הבאה: מימוש שעון באמצעות CPU, ניתן לביצוע על-ידי לולאות השהייה בתוכנית ה-CPU. לצורת מימוש זו קיימות שתי בעיות קשות. האחת, שה-CPU יכול לעסוק רק בתוכנית זו - בלולאות ההשהייה ועיבוד הנתון בסיום כל לולאה. בעיה אחרת היא, שזמן

עיבוד הנתון משתנה. לדוגמא, הזמן הדרוש לעדכון הדקות קצר מהזמן הנדרש לעדכון הדקות והשעות. בשימוש בקוצב זמן, מקבלים בקשת פסיקה כל פרק זמן קבוע, שניתן לתיכנות על-ידי המשתמש. ה-CPU יכול לעסוק בכל תוכנית שהיא. עם קבלת בקשת הפסיקה, הוא פונה לתוכנית הפסיקה ומעדכן את השעון בהתאם. מירווחי הזמן במקרה כזה יהיו אחידים.
דוגמאות אחרות לשימושים של המונים הן:

מונה אירועים.
חד יציב ספרתי.
מחולל גל ריבועי בעל תדר הניתן לתכנות.
יוצר צורות גל.
בקר מורכב למנוע זרם ישר.
בקר למנוע צעד.

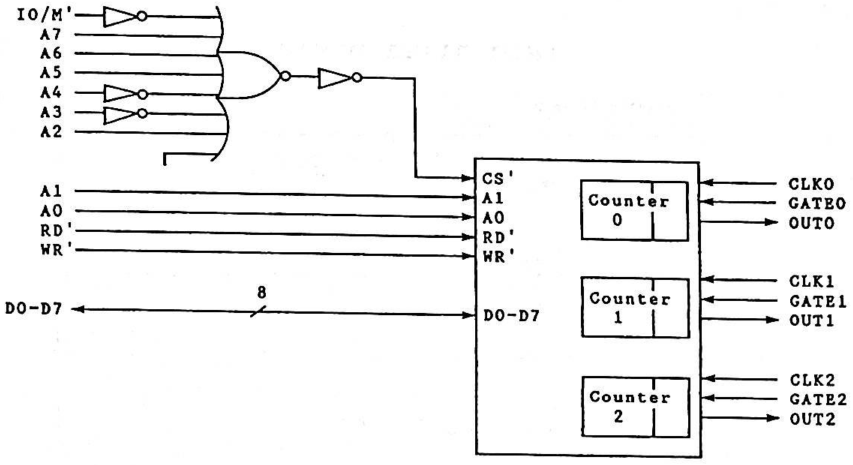
ל-8254 שלושה מונים לא תלויים זה בזה בני 16 סיביות כל אחד. דיאגרמת המלבנים של הרכיב ותיאור הפינים שלו מתוארים בציור הבא:



ציור 6-1 תיאור פינים ודיאגרמת מלבנים של ה-8254

הרכיב מיועד להתחבר ישירות ל-BUS ה-CPU ותופס ארבע כתובות I/O. לכן בנוסף לכניסת 'CS, המתקבלת מפענוח הקוים 'IO/M ו-A2-A7, מתחברים אליו הקוים A0,A1. הקוים 'RD ו-'WR מתחברים אליו ישירות לקריאה או לכתיבה.

נדגים חיבור הרכיב לתחום הכתובות IO-18H עד IO-1BH.

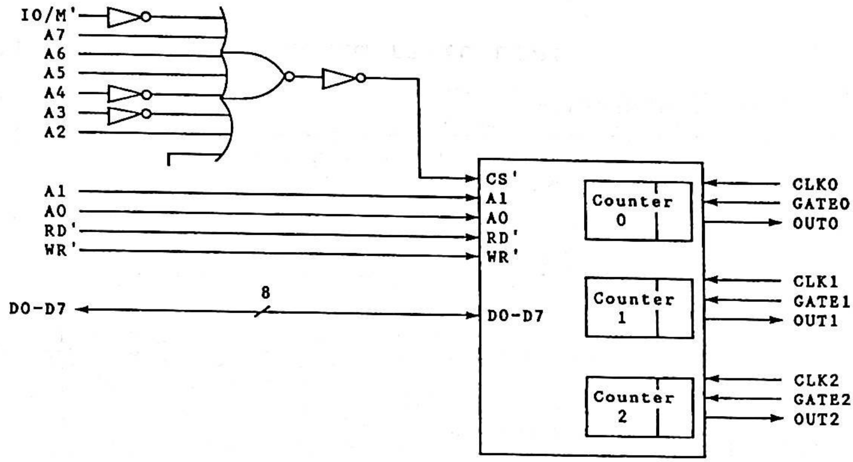


ציור 2-6 חיבור 8254 ל-BUS

שלושת המונים הם מונים עצמאיים לחלוטין. כל מונה הוא בן 16 סיביות ומורכב למעשה משני חלקים. חלק אחד כולל את סיביות ה-MSB והחלק השני את סיביות ה-LSB. לכל מונה כניסת CLK, דרכה הוא מקבל את כניסת השעון, כניסת שער שבעזרתה ניתן לחסום את אותות השעון ולהפסיק את הספירה והיציאה OUT. למונים ששה אופני עבודה שונים וכל מונה יכול לעבוד ב-MODE שונה. היציאה OUT משתנה בהתאם לאופן העבודה.

המונים הם מוני DOWN, כלומר, סופרים כלפי מטה. ניתן לטעון אותם בערך התחלתי וניתן לקרוא אותם. כדי לפנות למונה כלשהו, חייבים לפנות לפני זה לאוגר בקרה (Control Word). אוגר הבקרה תופס את הכתובת הגבוהה (A0,A1=11). המלה הנכתבת לאוגר הבקרה, קובעת את אופן העבודה של המונה וצורת הפניה לבתיו. ניתן לכתוב באוגר הבקרה שלוש מילות בקרה שונות עבור שלושת המונים. כל מלה כזו תישמר לכל מונה עד לכתיבה נוספת עבור אותו מונה. אם אין משנים את אופן העבודה וצורת הפניה במונה מסוים, אין צורך לפנות לאוגר הבקרה עבורו שוב, גם אם כותבים באוגר הבקרה

נדגים חיבור הרכיב לתחום הכתובות IO-18H עד IO-1BH.



ציור 6-2 חיבור 8254 ל-BUS

שלושת המונים הם מונים עצמאיים לחלוטין. כל מונה הוא בן 16 סיביות ומורכב למעשה משני חלקים. חלק אחד כולל את סיביות ה-MSB והחלק השני את סיביות ה-LSB. לכל מונה כניסת CLK, דרכה הוא מקבל את כניסת השעון, כניסת שער שבעזרתה ניתן לחסום את אותות השעון ולהפסיק את הספירה והיציאה OUT. למונים ששה אופני עבודה שונים וכל מונה יכול לעבוד ב-MODE שונה. היציאה OUT משתנה בהתאם לאופן העבודה.

המונים הם מוני DOWN, כלומר, סופרים כלפי מטה. ניתן לטעון אותם בערך התחלתי וניתן לקרוא אותם. כדי לפנות למונה כלשהו, חייבים לפנות לפני זה לאוגר בקרה (Control Word). אוגר הבקרה תופס את הכתובת הגבוהה (A0,A1=11). המלה הנכתבת לאוגר הבקרה, קובעת את אופן העבודה של המונה וצורת הפניה לבתיו. ניתן לכתוב באוגר הבקרה שלוש מילות בקרה שונות עבור שלושת המונים. כל מלה כזו תישמר לכל מונה עד לכתיבה נוספת עבור אותו מונה. אם אין משנים את אופן העבודה וצורת הפניה במונה מסוים, אין צורך לפנות לאוגר הבקרה עבורו שוב, גם אם כותבים באוגר הבקרה

שיכויים עבור המונים האחרים.

הכתיבה באוגר הבקרה מתוארת בציור הבא:

Control Word Format

A_1,A_0 = 11 $\overline{CS}$ = 0 $\overline{RD}$ = 1 $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC—Select Counter

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (see Read Operations)

M—Mode

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

RW—Read/Write

RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only
1	0	Read/Write most significant byte only
1	1	Read/Write least significant byte first, then most significant byte

BCD

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

ציור 6-3 Control Word Format

סיבית D0 (BCD) מציינת אם המונה סופר בארבע ספרות BCD (4 סיביות כל ספרה) או במספר בינארי בן 16 סיביות.

סיביות D1-D3 (M2-M0) קובעות את אופן עבודת המונה. אופני העבודה השונים יתוארו בהמשך.

סיביות D4,D5 (RW1,RW0) קובעות את אופן הפניה למונה, כפי שמתואר בטבלה RW1 RW0.

סיביות D6,D7 (SC1,SC0) קובעות לאיזה מונה מיועדת מלה בקרה זו.

עיין בטבלאות השונות שבציור 6-3.

לאחר כתיבת המלה, נקבעת מלת הבקרה עבור המונה בהתאם ל-SC1,SC0. כל מונה תופס כתובת אחרת. מונה 0 - כתובת

שבה A1,A0=00, מונה 1 - כתובת שבה A1,A0=01 ומונה 2 - כתובת שבה A1,A0=10.

לדוגמא, אם קבענו עבור מונה 1 (SC1,SC0=01) RW1,RW0=10, פירושו של דבר שכל פניה לכתובת מונה 1 (A1,A0=01) תפנה נתון לבית הגבוה שלו (במקרה של כתיבה) או תאסוף נתון מהבית הגבוה שלו (במקרה של קריאה). במקרה כזה אין אפשרות לפנות לבית הנמוך של המונה.

כדי לשנות את אופן הפניה למונה, שוב צריך לפנות לאוגר הבקרה עם SC1,SC0=01 ועם נתון חדש ב-RW1,RW0. אם למשל, נכתוב RW1,RW0=11, הפניה למונה היא לשני בתים. בפניה ראשונה למונה הנתון יגיע לבית הנמוך והפניה השניה תגיע לבית הגבוה. הפניה השלישית תגיע שוב לבית הנמוך וחוזר חלילה. אין זה משנה אם בין פניה לפניה נעשו פניות למונים אחרים.

מספר אפשרויות הפניה לאוגרי הבקרה ולמונים הוא רב ביותר. בציור הבא מתוארות מספר דוגמאות כדי להמחיש את הגמישות הרבה האפשרית.

	A_1	A_0		A_1	A_0
Control Word—Counter 0	1	1	Control Word—Counter 2	1	1
LSB of count—Counter 0	0	0	Control Word—Counter 1	1	1
MSB of count—Counter 0	0	0	Control Word—Counter 0	1	1
Control Word—Counter 1	1	1	LSB of count—Counter 2	1	0
LSB of count—Counter 1	0	1	MSB of count—Counter 2	1	0
MSB of count—Counter 1	0	1	LSB of count—Counter 1	0	1
Control Word—Counter 2	1	1	MSB of count—Counter 1	0	1
LSB of count—Counter 2	1	0	LSB of count—Counter 0	0	0
MSB of count—Counter 2	1	0	MSB of count—Counter 0	0	0

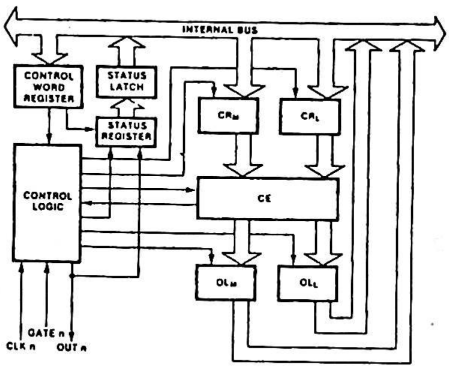
	A_1	A_0		A_1	A_0
Control Word—Counter 0	1	1	Control Word—Counter 1	1	1
Control Word—Counter 1	1	1	Control Word—Counter 0	1	1
Control Word—Counter 2	1	1	LSB of count—Counter 1	0	1
LSB of count—Counter 2	1	0	Control Word—Counter 2	1	1
LSB of count—Counter 1	0	1	LSB of count—Counter 0	0	0
LSB of count—Counter 0	0	0	MSB of count—Counter 1	0	1
MSB of count—Counter 0	0	0	LSB of count—Counter 2	1	0
MSB of count—Counter 1	0	1	MSB of count—Counter 0	0	0
MSB of count—Counter 2	1	0	MSB of count—Counter 2	1	0

NOTE:
In all four examples, all Counters are programmed to read/write two-byte counts. These are only four of many possible programming sequences.

ציור 4-6 מספר דוגמאות לתכנות ה-8254

באותה צורה שאנו כותבים במונה, כך אנו יכולים לקרוא ממנו. במקרה כזה עלינו לחסום את כניסת השעון למונה. לכל כניסת CLK למונה יש קו GATE. הורדת קו GATE ל-"0", עוצרת את המונה. עצירה זו חיונית בשלב הקריאה כי אם המונה יהיה במצב מעבר בזמן הקריאה ממנו, נקבל מספר שגוי.

במצב קריאה זה קיימת בעיה נוספת. נניח שמחובר למונה CLK מהיר. קריאת המונה נעשית בשני שלבים. ייתכן מצב שבין שתי הקריאות התבצעו מניות נוספות. כדי לפתור בעיות אלה, קיימות אפשרויות קריאה נוספות. האפשרות הראשונה, היא לתת למונה פקודת נעילה של המידע שבתוכו. כל מונה בנוי בצורה הבאה:



ציור 5-6 דיאגרמת מלבנים של מונה

בפקודת הנעילה מועבר המספר שנמצא במונה ברגע הפקודה לשני נועלים (Laches) האוגרים את המידע עד לפניית ה-CPU לרכיב לשם קריאת המידע. אם תגיע בקשת נעילה נוספת לפני שהתבצעה פעולת קריאה מהנועלים, לא יתייחס אליה הרכיב. פקודת הנעילה מתבצעת כאשר RW1,RW0=00 עבור מונה מסוים. מלת הבקרה במקרה זה היא:

SC1,SC0,0,0,X,X,X,X

קימת פעולת נעילה ואגירת מידע מורכבת יותר. פקודה זו מתבצעת כאשר SC1,SC0=11. בפקודה כזו מציינים איזה מידע אנו מעוניינים לנעול. זאת באמצעות פקודת הבקרה הבאה, הנקראת Read Back Command.

A0, A1 = 11 $\overline{CS}$ = 0 $\overline{RD}$ = 1 $\overline{WR}$ = 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	$\overline{COUNT}$	$\overline{STATUS}$	CNT 2	CNT 1	CNT 0	0

D_5: 0 = Latch count of selected counter(s)
D_4: 0 = Latch status of selected counters(s)
D_3: 1 = Select Counter 2
D_2: 1 = Select Counter 1
D_1: 1 = Select Counter 0
D_0: Reserved for future expansion; Must be 0

ציור 6-6 פקודת Read Back Command

בפקודה זו ניתן לבקש נעילת תוכן כל אחד מהמונים (כל צירוף אפשרי), או מצב (STATUS) של אופן עבודתם. כאשר מבקשים נעילת מונה צריכה סיבית count להיות ב-"0", והסיבית המתאימה למונה המבוקש להיות ב-"1". בבקשת נעילת מצב המונה צריכה סיבית status להיות ב-"0". המידע ישאר נעול עד לקריאתו על-ידי ה-CPU. כאשר קוראים את STATUS המונה, מתקבל המידע הבא:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Output	Null Count	RW1	RW0	M2	M1	M0	BCD

D_7 1 = OUT Pin is 1
0 = OUT Pin is 0
D_6 1 = Null Count
0 = Count available for reading
D_5–D_0 Counter programmed mode (see Figure 7)

ציור 6-7 Status byte

תיתכן בקשת נעילת מידע של Sstatus ו-count יחדיו. במקרה כזה יש לפנות למונה 3 פעמים כדי לקרוא את המידע. המידע הראשון שיגיע יהיה ה-Status byte.

כאמור, למונים של ה-8254 ששה אופני עבודה. להלן הסבר תמציתי על אופני עבודה אלה. ציור 8-6 מתאר דיאגרמות זמנים שונות עבור אופני עבודה אלה.

MODE0

ב-MODE0 מתפקד המונה כמונה אירועים. לאחר קביעת המונה לאופן עבודה זה, טוענים את המונה במספר המניות הנדרש. אז מתחיל (בתלות ב-GATE) המונה לספור down את דפקי ה-CLK המסופקים אליו. ברגע שהוא מגיע ל-0, קו OUT עולה ל-High. קו זה יכול לשמש כקו פסיקה.

MODE1

ב-MODE1 מתפקד המונה כחד יציב. קו ה-GATE משמש כקו Trigger וקו OUT יורד ל-Low למשך אורך המספר שנרשם במונה, לאחר תכנותו לאופן עבודה זה.

MODE2

ב-MODE2 משמש המונה כמחולל קצב שידור. לאחר תכנות המונה,הוא מתפקד כמחולל דפקים היוצר דופק שלילי במשך זמן מחזור שעון אחד כל מספר דפקי שעון, בהתאם למספר שנרשם במונה. ניתן לחסום את יציאת המחולל (קו OUT) בעזרת קו GATE. ניתן לשנות את קצב הדפקים תוך כדי עבודתו, בכתיבה אחת לתוכו.

MODE3

MODE3 דומה ל-MODE 2 מלבד העובדה שהגל שנוצר הוא גל ריבועי. קו OUT נמצא ב-High למשך מחצית מספר דפקי השעון הרשומים במונה וב-Low למשך המחצית השניה. כלומר, מספר דפקי השעון מציינים את זמן המחזור. קו GATE מאפשר לחסום

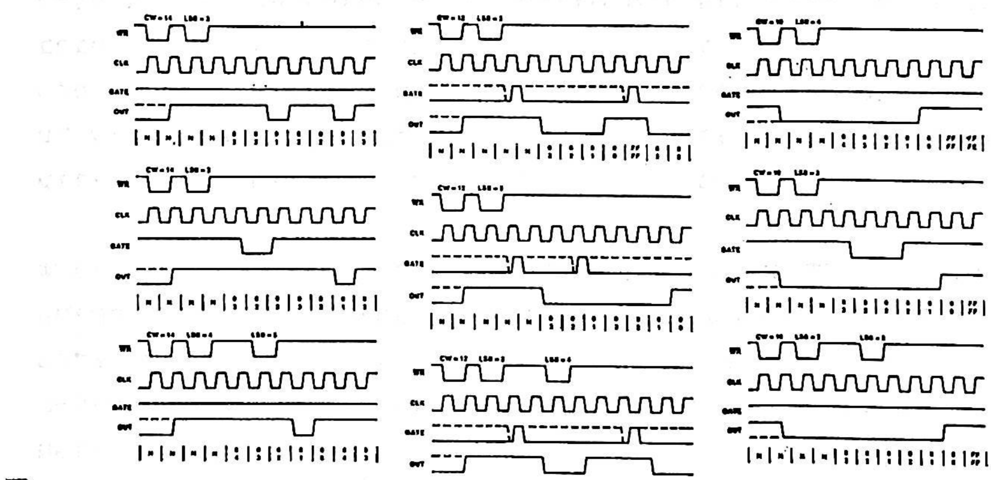
את הגל הריבועי. ניתן לשנות את תדירות הגל תוך כדי פעולת המונה.

MODE4

MODE4 מיועד ליצור אות strobe לאחר השהייה הנקבעת בתוכנה. לאחר תכנות המונה לאופן עבודה זה, קו OUT נמצא ב-High עד לסיום המניה. ברגע שהסתיימה המניה, יורד קו OUT ל-Low באופן חד פעמי למשך מחזור שעון אחד. קו GATE חוסם את פעולת המניה, ומאריך את ההשהייה.

MODE5

MODE5 הוא שילוב של MODE 1 ו-MODE 4. לאחר תכנות המונה לאופן עבודה זה, קו OUT נמצא ב-High עד לקבלת אות עירור דרך קו GATE. אז נטען המונה ומתחילה המניה לאחור. ברגע שמסתיימת המנייה יורד קו OUT ל-Low למשך דופק שעון אחד. ברגע שיגיע אות עירור נוסף תתחיל מנייה נוספת.



MODE2 MODE1 MODE0

ציור 8-6 דיאגרמות זמנים של אופני עבודת המונים

למונה/קוצב זמן יש מספר רב של יישומים, וחשיבותו במערכות משובצות מיקרופרוססור גדולה. הרכיב מפשט בצורה משמעותית את תוכנת המערכת. בסעיפים הבאים מתוארות מספר דוגמאות מתוך המגוון הרחב האפשרי.

6.2 קוצב זמן, מד תדר, Watch dog

כאשר מחברים למונה אות בעל תדירות קבועה, מתפקד המונה כקוצב זמן. מפעילים את המונה ב-MODE0 (ראה דיאגרמת זמנים). טוענים אותו במספר מחזורי השעון, שרוצים לקבל כהשהייה. המונה יספור DOWN. המניה תסתיים כאשר המונה יגיע ל-0, וקו OUT יעלה ל-"1". עליה זו מסמנת את סיום ההשהייה.

ניתן לדגום את המונה ב-Polling. אפשרות אחת היא לקרוא את המונה ולבדוק מתי הוא מגיע ל-0. האפשרות שניה היא לחבר את קו OUT לפורט מבוא ולבדוק מתי הוא עולה ל-"1". ברגע שמגלים את סיום המניה, טוענים אותו שוב, מעבדים את הזמן וחוזרים לדגום את המונה. בכל מקרה עדיפה שיטה זו על לולאת השהייה. זמן ההשהיות יהיה אחיד ולא תלוי בזמן עיבוד ההשהייה (בהנחה שזמן העיבוד קצר מזמן ההשהייה).

שיפור נוסף, שניתן לעשות, הוא לחבר את קו OUT לכניסת פסיקה של ה-CPU. תוכנית הפסיקה, מבצעת את עדכון המונה מחדש ואת עיבוד הנתון. בחזרה מתוכנית הפסיקה, משוחרר ה-CPU לפעולות שונות ואינו צריך להיכנס ללולאת דגימה של המונה.

מספר דוגמאות לשימוש המונה כקוצב זמן. נניח שצריכים לאסוף מידע מפורט מבוא, כל פרק זמן מוגדר. קוצב הזמן מכתיב את קצב איסוף האינפורמציה. המידע יכול להגיע גם

ADC-מ (Analog to Digital Converter - ממיר מתקבלוי לספרתי). בצורה כזו דוגמים כל פרק זמן את האות האנלוגי, וממירים אותו לספרתי. ניתן גם להציג את המידע בתצוגה ספרתית. ראינו בפרק 4 תצוגה בריבוב. ניתן לחבר את יציאת קוצב הזמן לקו פסיקה. בכל פסיקה של הקוצב, מעבירה תוכנית הפסיקה את התצוגה ליחידת תצוגה אחרת. אותו הדבר לגבי דגימת לוח מקשים.

נמחיש את השימוש בקוצב הזמן בעזרת דוגמא משולבת. נחבר יציאת קוצב זמן לקו בקשת פסיקה אחד ויציאת קוצב זמן אחר לקו בקשת פסיקה שני. נחבר ל-ADC CPU, הממיר מתח למספר בינארי, ותצוגת 7-SEG. בריבוב. בכל בקשת פסיקה של Timer1, דוגם ה-CPU את ה-ADC וממיר את המספר הנדגם לשלוש ספרות BCD. את הספרות מאחסן בשלושה תאי זכרון. בכל בקשת פסיקה של Timer1, מציג ה-CPU תוכן של תא זכרון אחד (מבין שלושת התאים) ביחידת תצוגה מתאימה. ה-CPU יכול לפעול על תוכנית ראשית כלשהי ולמרות זאת יופיע כל הזמן בתצוגה הערך של האות האנלוגי שנמצא בכניסת ה-ADC. ניתן להוסיף בצורה כזו גם טיפול בלוח מקשים וכד'.

שני קוצבי הזמן יכולים להיות מוזנים מאותו מחולל גל ריבועי ולמרות זאת לתת פסיקות בזמנים שונים, על-ידי היתחול שונה לכל אחד מהם.

שימוש אחר של המונה/קוצב הזמן הוא למדידת תדר. ניתן להשתמש בשני מונים. לאחד מספקים גל ריבועי שאת תדרו רוצים למדוד. המונה האחר ישמש כקוצב זמן. מאפסים את המונה ומהתחלים את קוצב הזמן. ברגע שקוצב הזמן מסמן סיום של פרק זמן, אוספים את המספר מהמונה. אם המונה סופר DOWN, לוקחים את המשלים ל-2 שלו. את המספר, המציין מספר מחזורים ביחידת זמן, מוציאים לתצוגה. מאפסים שוב את המונה ומהתחלים שוב את קוצב הזמן וחוזר חלילה. גם פעולות אלה יכולות להתבצע בעזרת בקשות פסיקה ובכך לשחרר

את ה-CPU לטיפול בתוכנית ראשית כלשהי.

שימוש נוסף של קוצב הזמן נקרא כלב שמירה (Watch dog). למערכת בקרה משובצת מיקרופרוססור, קיימת הבעיה הבאה. ייתכן, שכתוצאה מרעש כלשהו, הפסקת חשמל רגעית, תכנון שלא הביא בחשבון את כל האפשרויות, תיקלע המערכת למצבים שלא נחזו מראש. ה-CPU יוצא מהתוכנית ומתחיל לבצע דברים לא מוגדרים. אם עומד אדם ליד המכשיר ויכול להבחין במצב כזה, הוא פשוט לוחץ RST. אבל, אם המערכת מתפקדת בצורה עצמאית, זה יכול לעתים להיות מסוכן ובודאי שזה מצב לא רצוי.

כדי לזהות מצב כזה, מוסיפים למערכת קוצב זמן, שיציאתו מחוברת לקו RST של המערכת. מהתחלים את קוצב הזמן. במהלך התוכנית הראשית, פונים מדי פעם לקוצב הזמן וטוענים אותו מחדש. דואגים שלעולם לא יגיע לקצה הספירה ויתן RST למערכת. אם ה-CPU יוצא מהתוכנית הרצויה, הוא יפסיק לטעון את קוצב הזמן ובגמר ספירתו, הוא יבצע RST למערכת באופן אוטומאטי. זו הסיבה, שמערכת כזו נקראת Watch dog. ניתן להשתמש בקוצב זמן שניתן לתיכנות, אם כי בדרך כלל משתמשים בקוצב זמן נפרד, שלא תלוי ב-CPU.

6.3 <u>מחולל גל מלבני, בקר למנועים</u>

קוצב הזמן יכול לשמש את ה-CPU כמחולל גל מלבני או ריבועי, ושוב מבלי להפריע ל-CPU, היכול לרוץ על התוכנית הראשית. בעזרת ה-8254 הדבר פשוט במיוחד. הפעלתו ב-MODE3 יוצרת גל ריבועי שזמן המחזור שלו ניתן לקביעה. כדי לקבל גל מלבני, שניתן לתכנת גם את זמן המחזור וגם את משך הדופק (Duty Cycle) שלו, אפשר להשתמש בשני קוצבי זמן. את האחד נפעיל ב-MODE2 כך שכל משך זמן המחזור הרצוי יתן

אות Strobe. אות זה נחבר לכניסת ה-GATE של קוצב זמן שני, שנפעילו ב-MODE1 (חד יציב). יציאת הקוצב השני תהווה יציאת המחולל. תכנות קוצב הזמן השני קובע את משך הדופק. עבודת הקוצבים היא עצמאית, כל זמן שאין צורך לבצע שינוי, הם פועלים מבלי להפריע ל-CPU.

גל מלבני, הניתן לתיכנות, יכול לשמש למעגלי בקרה שונים. אם נספק גל כזה למנוע זרם ישר, יקבע ה-Duty Cycle את מהירות המנוע. Duty Cycle מוגדר כיחס בין משך הדופק לזמן המחזור באחוזים.

$$\text{Duty Cycle} = \frac{ton}{T} * 100$$

בצורה כזו נוכל לשלוט על מהירות של מנוע DC בתוכנה. נצטרך כמובן מעגל דחיפה כלשהו, כדי לספק את הזרם הנדרש למנוע. ניתן לחבר לציר המנוע יחידה המספקת כל סיבוב דופק מתח. נמדוד את תדר הדפקים בעזרת מונה נוסף ונשווה עם המהירות הרצויה. אם המהירות שונה, נוכל לשנותה בהתאם.

בעזרת קוצב הזמן ניתן לבקר על מהירות מנוע צעד (הסבר על מנוע DC ומנוע צעד נמצא בספר "יישומי מחשב באסמבלי" בהוצאת אס.אי.אס. מערכות לימוד). מפעילים קוצב זמן למשך ההשהייה הנדרשת בין הצעדים. בכל בקשת פסיקה המתקבלת, מקדמים את המנוע בצעד אחד.

6.4 שעון זמן אמיתי

שעון זמן אמיתי הוא למעשה קוצב זמן, שתואר בסעיף 6.2. תוכנית הפסיקה מבצעת, בדרך כלל, מניה ביותר מתא אחד. שעון העובד על בסיס השעון הרגיל, צריך להתחשב בחלוקה של ששים שניות בדקה, ששים דקות בשעה ועשרים וארבע שעות ביממה. למערכת כזו מקציבים מספר תאים לאיחסון הערכים השונים. תוכנית הפסיקה, דואגת להוסיף כל פסיקה לתאים את הערך הדרוש. תוכנית כזו מראה את הצורך בפסיקה חיצונית. הזמן הדרוש לחישוב שונה מחישוב לחישוב. זמן החישוב, כאשר יש נשא מספרה לספרה, ארוך יותר מכאשר התוכנית מעלה רק את הספרה הפחות משמעותית.

קיימים רכיבים המבצעים את מניית הזמן בצורה עצמאית לחלוטין, כולל עדכון שעות ותאריך. הם משחררים את ה-CPU לחלוטין. הפניה אליהם יכולה להיות כפניה אל פורט, או במיפוי זכרון. ניתן להגן עליהם בעזרת סוללה ולקבל שעון זמן אמיתי ממש.

6.5 שאלות ותרגילים

6.5.1 שרטט חיבור 8254 לתחום הכתובות IO-18H עד IO-1BH. השתמש במפענח 74138.

6.5.2 לכניסת מונה 0 מחובר גל ריבועי בתדר 6.144MHz. רגל ה-GATE שלו מחוברת ל-VCC. נדרש גל ריבועי בתדר 76,800Hz ביציאת OUT0. רשום פקודות היתחול ל-8254 לקבלת אות כזה.

6.5.3 לכניסת מונה 1 מחברים מחולל גל ריבועי בתדר 50Hz. לקו GATE1 מחובר מפסק. ליציאת OUT1 מחובר LED (דרך דוחף), כך שירידת OUT1 ל-"0" מכבה

אותו.

רשום תוכנית לקביעת מונה 1 כחד יציב למשך 5 שניות. כלומר, לאחר היתחול הרכיב, העלאת המפסק ל-"1" והורדתו ל-"0", תגרום לכיבוי ה-LED למשך 5 שניות.

6.5.4 לכניסת מונה 1 מחברים גל ריבועי בתדר קבוע של 50Hz. יציאת מונה זה מחברים לקו בקשת פסיקה. לכניסת מונה 2 מחברים גל בתדר משתנה.

רשום תוכנית ראשית הגורמת למונה 1 לעבוד ב-MODE2, כך שיצור בקשת פסיקה כל 0.1spc. מונה 2 יתפקד כמונה אירועים ב-MODE0. התוכנית מאפסת אותו. רשום תוכנית פסיקה, הקוראת את תוכן מונה 2, מבצעת משלים ל-2 שלו ומוציאה אותו לנוריות בכתובת IO-00.

6.5.5 למונה 1 ולמונה 2 מחובר מחולל גל ריבועי בתדר 50Hz. את יציאת מונה 1 מחברים דרך מהפך ל-GATE של מונה 2. את יציאת מונה 2 מחברים דרך מהפך לדוחף המפעיל מנוע DC.

רשום פקודות היתחול הגורמות למונה 1 לתפקד ב-MODE2, כך שיצור אות עירור למונה 2 כל שניה אחת.

מונה 2 מתפקד כחד יציב בעל אורך דופק של 0.4 שניות. המנוע צריך לפעול כל שניה אחת למשך 0.4 שניות.

תרגיל

6.6 יצירת גל ריבועי

מטרת התרגיל ליצור גל ריבועי בעזרת קוצב הזמן.

כמתואר בשאלה 6.5.2, מחובר לכניסת מונה 0 גל ריבועי בתדר 6.144MHz. יציאת המונה מתחברת לכניסת ה-CLK של ה-USART, המתואר בפרק 8.

6.6.1 רשום תוכנית המבצעת היתחול ל-8254 כך, שיצור גל בתדירות של 76,800Hz ביציאת OUT0. הסיבה לתדירות זו תובהר בפרק 8.
התוכנית מסתיימת בחזרה לתוכנית המוניטור.

6.6.2 הרץ את התוכנית.

6.6.3 חבר משקף תנודות לרגל OUT0 של ה-8254 (רגל 10) ובדוק את האות ותדירותו.

6.6.4 שים לב, שכל זמן שאינך פונה למונה 0, מופיע האות ביציאת הרכיב.

6.6.5 שנה את פקודות ההיתחול כך שהתדירות תוכפל. הרץ את התוכנית ובדוק שוב.

6.7 מדידת זמן מחזור של מתנד משתנה

מטרת התרגיל ליצור מד תדר בעזרת ה-8254.

6.7.1 חבר לכניסת מונה 1 (CLK1) גל ריבועי מהמתנד הקבוע 50Hz הנמצא בערכה S-2100.

6.7.2 חבר את GATE1 ל-VCC.

6.7.3 חבר את OUT1 לקו בקשת פסיקה. ב-S-2051 לקו INT1 וב-S-2085 לקו RST6.5.

6.7.4 חבר את יציאת המתנד המשתנה, הנמצא בערכה S-2100, לכניסת מונה 2 (CLK2).

6.7.5 רשום תוכנית ראשית המבצעת:

א) חסימת בקשות פסיקה.

ב) מפעילה את מונה 1 לעבודה ב-MODE2 כך שיצור בקשת פסיקה של 0.1sec (חלוקה ב-5).

ג) מפעילה את מונה 2 לעבודה ב-MODE0 כמונה אירועים.

ד) מאפסת את מונה 2.

ה) מבצעת היתחול ל-8255 להפעלת פורט A כפורט מוצא.

ו) מאפשרת בקשת פסיקה.

ז) מדלגת לתוכנית "אור רץ" שבערכה (ב-S-2051 כתובת 0036 וב-S-2085 בכתובת 04E9).

6.7.6 רשום תוכנית פסיקה המבצעת:

א) קריאת תוכן מונה 2 לצובר.

ב) משלים ל-2 של תוכן הצובר.

ג) הוצאת תוכן הצובר לפורט A של ה-8255.

ד) מאפסת את מונה 2.

ה) מאפשרת פסיקה.

ו) חזרה מתוכנית פסיקה.

6.7.7 ודא שמעגל נוריות מפסקים (DSM-2150) מחובר ל-8255.

6.7.8 S-2051 ו-S-2085 כוללים כתובת ב-RAM, ששם יש לכתוב פקודת דילוג לתוכנית הפסיקה.

ב-S-2051, הכתובת לבקשת פסיקה INT1 היא 4003H.

ב-S-2085, הכתובת לבקשת פסיקה RST6.5 היא 38BEH.

רשום את הפקודה המתאימה בכתובת זו בהתאם לערכה שברשותך.

6.7.9 הרץ את התוכנית הראשית.

בנוריות הערכה צריך להופיע "אור רץ".

בנוריות המחוברות ל-8255 צריך להופיע מספר בינארי כפונקציה של תדר המתנד המשתנה.
שנה את תדר המתנד המשתנה ובדוק שהמספר משתנה.
מדוד את תדירותו בעזרת מד תדר או משקף תנודות והשווה עם המספר שמתקבל.

6.7.10 אם המערכת לא מתפקדת, בדוק:

א) שמתקבלות בקשות פסיקה. כלומר, לקו בקשת הפסיקה מגיע דופק של 0.1sec. בדוק זאת בעזרת משקף תנודות.

ב) רשום תוכנית פסיקה המוציאה FF לנוריות הערכה ועוצרת. הרץ את התוכנית ובדוק שהנוריות נדלקו ברגע שהגיעה בקשת פסיקה. זה צריך לקרות מיד.

ג) אם סעיפים א' ו-ב' תקינים, בדוק את תוכנית הפסיקה שלך.

6.8 בקרת מנוע

מטרת התרגיל להפעיל מנוע DC במהירויות שונות בהתאם למצב מפסקים.

6.8.1 חבר יציאת מחולל גל ריבועי בתדירות 50Hz לכניסות המונים 1 ו-2.

6.8.2 חבר יציאת מונה 1 דרך מהפך ל-GATE של מונה 2.

6.8.3 חבר יציאת מונה 2 לכניסת דוחף שמפעיל LED (4).

6.8.4 וודא חיבור מעגל מפסקים נוריות (DSM-2150) ל-8255.

6.8.5 רשום תוכנית המבצעת:

א) היתחול ל-8255 כך שפורט C ישמש כפורט מבוא.

ב) היתחול ל-8254 כך שמונה 1 יתפקד ב-MOSE2 ומונה 2 יתפקד כחד יציב.

ג) מונה 1 יתן דופק עירור למונה 2 כל שניה אחת.

ד) קריאת מצב המפסקים מפורט C של ה-8255 לצובר.

ה) איחסון תוכן הצובר במונה 2, כך שאורך דופק החד יציב יהיה בהתאם למצב המפסקים.

ו) דילוג לסעיף ד'.

6.8.6 הרץ את התוכנית.

ה-LED צריך להבהב אחת לשניה כאשר משך ההדלקה תלוי במצב המפסקים.

6.8.7 אם תחבר מנוע DC ליציאת הדוחף המפעיל את ה-LED, מהירותו תשתנה בהתאם למצב המפסקים.

A to D-ו D to A

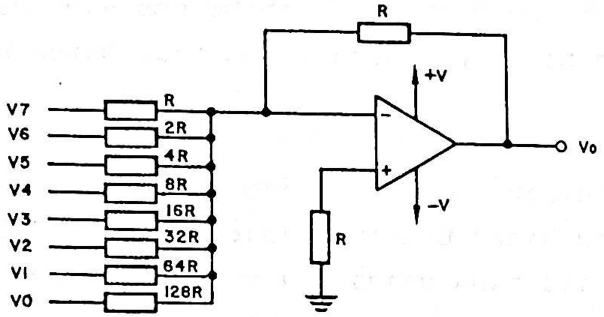
7.1 DAC בעזרת מגבר שרת ורשת נגדים

מעגל Digital to Analog Converter הוא מעגל המקבל בכניסותיו מספר בינארי כלשהו ומוציא ביציאותיו מתח חשמלי. המספר הבינארי מורכב מספרות בינאריות (סיביות) שיכולות לקבל ערך לוגי "0" או "1", כאשר "0" שווה ל-0V ו-"1" שווה למתח ייחוס מסוים - VR.

המתח במוצא המעגל שווה לערך העשרוני של המספר הבינארי מוכפל ב-VR ומוכפל במקדם, שהוא פונקציה של המעגל. לדוגמא, נניח שהמספר הבינארי הוא 01001100 השווה ל-78_{10}, מתח הייחוס הוא 5V ומקדם המעגל הוא 0.01. במוצא המעגל יתקבל המתח:

VO = 0.01 * 5 * 78 = 0.05 * 78 = 3.9V

בניית מעגל DAC נעשית בדרך כלל בעזרת מגבר שרת + רשת נגדים. בשיטה זו בונים למעשה מגבר מסכם כמתואר בציור 7-1.



ציור 7-1 מעגל DAC בעזרת מגבר שרת ורשת נגדים בעלי שקלול בינארי

נחשב את מתח המוצא V0 כפונקציה של V0-V7.

$$V0 = \frac{R}{R} V7 + \frac{R}{2R} V6 + \frac{R}{4R} V5 + \frac{R}{8R} V4 + \frac{R}{16R} V3 + \frac{R}{34R} V2 + \frac{R}{64R} V1 + \frac{R}{128R} V0$$

נצמצם ב-R

$$V0 = (V7 + \frac{1}{2} V6 + \frac{1}{4} V5 + \frac{1}{8} V4 + \frac{1}{16} V3 + \frac{1}{32} V2 + \frac{1}{64} V1 + \frac{1}{128} V0)$$

נכפיל ונחלק ב-128

$$V0 = \frac{1}{128} (128V7 + 64V6 + 32V5 + 16V4 + 8V3 + 4V2 + 2V1 + V0)$$

אם כל כניסת מתח שווה למתח הייחוס VR כפול ערך הסיבית המחוברת אליה ("0 " או "1")

VI = VR * DI

אזי נקבל

$$V0 = \frac{1}{128} (128VR*D7+64VR*D6+32VR*D5+16VR*D4+8VR*D3+4VR*D2+2VR*D1+VR*D0)$$

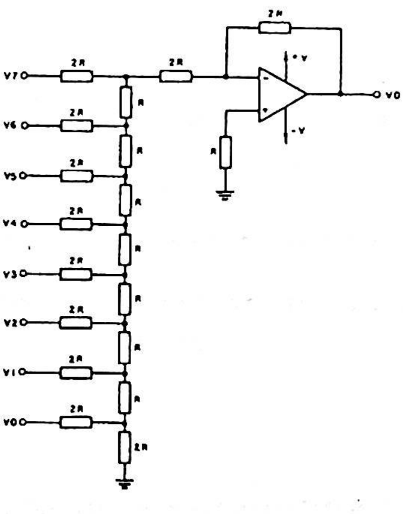
$$V0 = \frac{1}{128} VR (128D7 + 64D6 + 32D5 + 16D4 + 8D3 + 4D2 + 2D1 + 1D0)$$

בצורה כזו מתקבל מתח המוצא שערכו המוחלט הוא פונקציה של המספר הבינארי, הנמסר לכניסות המעגל. המקדם במעגל זה שווה לאחד חלקי 128.

המתח במוצא הוא מתח שלילי. כדי לקבל מתח חיובי ניתן לחבר למוצא המעגל מגבר מהפך. ניתן לתת לו גם הגבר מסוים כדי לשנות את המקדם הכללי של המעגל.

במעגל DAC חשיבות מרובה לדיוק הנגדים. הבעיה במעגל זה היא השימוש ברשת נגדים שונים החייבים לשמור על ערך יחסי מדויק זה מזה, כמתואר בציור. בניית מעגל מדויק כזה היא קשה ויקרה יחסית, אם כי קיימות רשתות נגדים מדויקות במארז אחד.

פתרון לבעיה זו ניתן למצוא במעגל המתואר בציור 7-2. מעגל זה משתמש ברשת של נגדים זהים (2 סוגים בלבד). רשת זו מכונה רשת סולם.



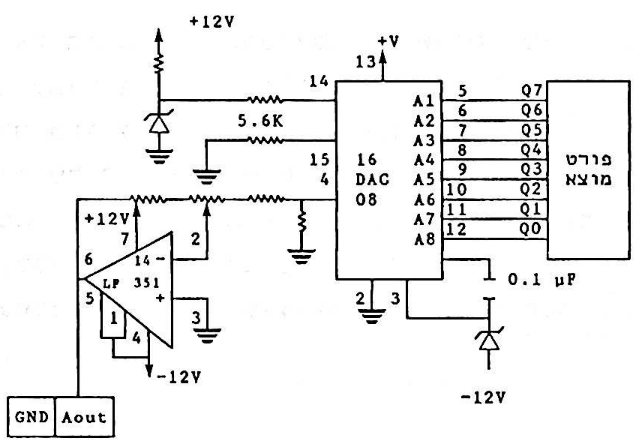
ציור 7-2 מעגל DAC עם מגבר שרת ורשת נגדים מסוג סולם של R ו-2R

מצא את הקשר בין מתח המוצא למתחי המבוא. לנוחיותך השתמש בשיטת הסופרפוזיציה.

7.2 DAC מונוליתי - DAC 08

DAC מונוליתי הוא ממיר מספרתי לתקבילי הבנוי בפיסה אחת. כלומר, מעגל משולב הכולל את רשת הנגדים, מגבר השרת ומעגלי קיזוז שונים. קיימים סוגים רבים של DAC, שההבדלים העיקריים ביניהם הם ברזולוציה שלהם (מספר סיביות המספר הבינארי), מהירות התגובה והדיוק. בדרך כלל מוציאים הרכיבים זרם שהוא פונקציה של המספר הבינארי בכניסותיו. בעזרת מגבר שרת ממירים זרם זה למתח בתחום המבוקש. את כניסות ה-DAC מחברים ליציאות של פורט מוצא.

נתאר חיבור של DAC פופולארי הנקרא DAC 08:

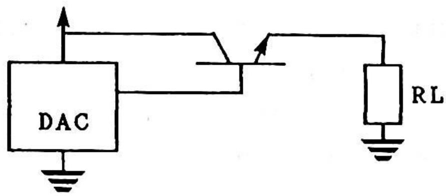


ציור 7-3 מעגל DAC ל-0-5V באמצעות DAC 08

כניסות A1-A8 משמשות ככניסות המספר הבינארי. שים לב ש-A1 הוא ה-MSB ו-A8 הוא ה-LSB (שמות הרגליים נקבעו על-ידי היצרן). רגל 3 משמשת ככניסת מתח שלילי VEE ורגל 16 ככניסה לקבל קיזוז. רגל 14 משמשת ככניסת מתח ייחוס חיובי +VREF ורגל 15 ככניסת מתח ייחוס שלילי -VREF. זרם המוצא של ה-DAC, המופיע ברגל 4, הוא פונקציה של הפרש הפוטנציאלים בין כניסות רגליים 14 ו-15 והמספר הבינארי

שבכניסותיו. הפוטנציומטר, שבכניסת מגבר השרת, משמש לכיול הגברו לקבלת התחום הרצוי. בדרך כלל, נחבר את רגל 15 לאדמה ולרגל 14 נספק מתח ייחוס מסוכן ומיוצב.

כאשר רוצים להפעיל רכיב כלשהו, בדרך כלל נדרש זרם גבוה יותר ממה שמגבר השרת יכול לספק. במקרה כזה מוסיפים מגבר זרם בצורה הבאה:



ציור 7-4 DAC עם מגבר זרם

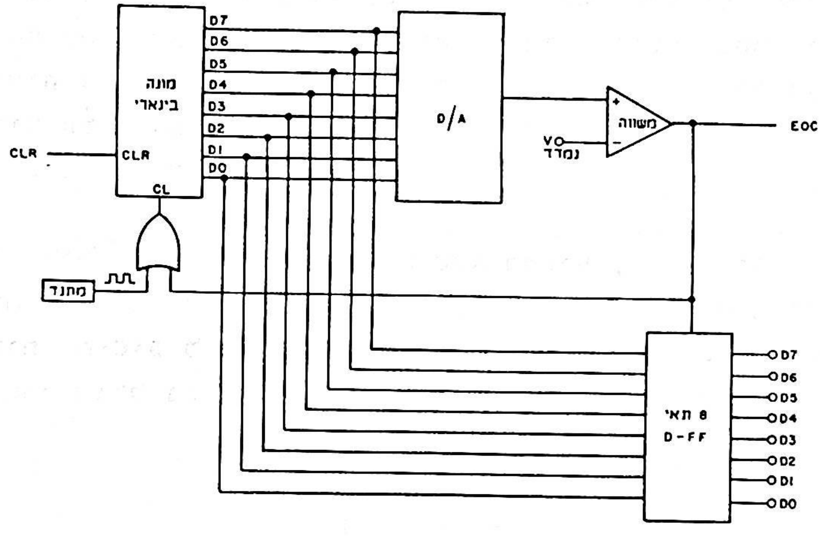
ה-DAC מספק את הזרם IB לטרנזיסטור השווה ל- . במידת הצורך ניתן להוסיף מגבר דרלינגטון כדי לקבל הגברת זרם גבוהה. בצורה כזו ניתן לממש גם ספק כח מבוקר.
שימוש מעניין של DAC הוא שימושו כמגבר מבוקר μP גם לזרם חילופין. מחברים לכניסת VREF מתח חילופין המוגבר לערך המכסימלי (ללא עיוותים כמובן). יציאת ה-DAC תהיה מתח חילופין שעוצמתו היא פונקציה של המספר הבינארי שבכניסותיו.

7.3 ADC בעזרת DAC

מעגל ADC (Analog to Digital Converter) הוא מעגל בעל כניסת מתח אנלוגי וקוי מוצא בינאריים. מעגל זה מוציא בקוי המוצא שלו את הערך הבינארי של מתח הנמדד שבכניסתו כפונקציה של מתח ייחוס מסוים.

מעגל ADC כולל בדרך כלל, מגבר משווה, שלכניסה אחת שלו מחובר המתח הנמדד. לכניסה השניה מספקים מתח הגדל בצורה הדרגתית כפונקציה של מספר בינארי גדל. ברגע שאות זה גדול מהמתח הנמדד, מחליף המגבר המשווה מצב ועוצר את העלאת המתח. שינוי זה גם גורם להוצאת המספר הבינארי שהתקבל ליציאות ה-ADC.

אחת השיטות לקבלת את הגדל בצורה הדרגתית היא שיטה הנעזרת ב-DAC שתואר בסעיף הקודם. ראה ציור 5-7.



ציור 5-7 מעגל ADC בעזרת DAC

ההמרה מתחילה במתן פולס CLR למונה. איפוס המונה גורם להוצאת 0V ביציאת ה-DAC שגורם לירידת קו EOC ל-LOW. ירידה זו מאפשרת העברת פולסים למונה שמתחיל למנות.

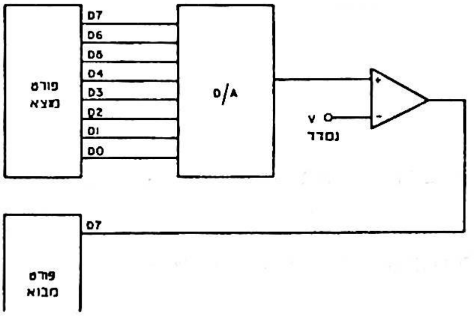
המספרים הבינאריים הגדלים שביציאות המונה, גורמים להופעת מתח גדל בצורה הדרגתית בכניסה הלא מהפכת של המגבר המשווה. ברגע שמתח זה עובר את המתח הנמדד, עולה קו EOC ל-HIGH.

עליה זו עוצרת את העברת הפולסים למונה ומסיימת את הספירה, וכן גורמת להכנסת המספר הבינארי שביציאות המונה לתאי ה-D-F.F. קו EOC משמש גם כקו המסמן למערכת המתחברת ל-ADC שההמרה הסתיימה (End Of Conversion).

אם רוצים לחבר ADC כזה למיקרו-מחשב יש לחבר את יציאותיו לפורט מבוא או לחוצץ שיצא מ-3-STATE רק עם פניית ה- CPU אליו.

אנו רואים שהמרה מתקבילי לספרתי מתרחשת במשך זמן מסוים. זו הסיבה שלכל מערכת ADC יש מלבד הכניסת מתח כמדד ויציאות בינאריות, קו כניסה המתחיל את ההמרה וקו יציאה המודיע על סיום תהליך ההמרה. אין לדגום את יציאות ה-ADC לפני תום תהליך ההמרה.

ניתן לותר על חלק ממרכיבי ה-ADC המתואר, על-ידי המרת חומרה בתוכנה. במקום לחבר מונה, מתנד ותאי FF מחברים את כניסות ה-DAC ליציאותיו של פורט מבוא ויוצרים את המספר הבינארי הגדל בתוכנה. מעגל זה יראה בצורה הבאה:



ציור 7-6 ADC באמצעות DAC ותוכנה

מוציאים לפורט המוצא מספרים בינאריים גדלים בזה אחר זה, המתחילים מ-0. לאחר הוצאת כל מספר לפורט בודקים את

סיבית D7 בפורט המבוא. ברגע שמוצאים שסיבית זו עלתה ל-"1", מפסיקים את הספירה. המספר האחרון שנשלח הוא הערך הבינארי המבוקש.

צריך להזכיר שני מושגים נוספים, הקשורים ב-ADC. מושג אחד נקרא Successive Approximation. ב-ADC הפועל בשיטת המניה, המרת מתח גבוה לוקחת זמן ארוך יותר מאשר המרת מתח נמוך. קיימת שיטה הנקראת Seccessive Approximation שבה מחברים במקום מונה, אוגר הזזה (Shift Reg.). מכניסים "1" לסיבית ה-MSB ומשווים את מתח היציאה ב-DAC למתח הנמדד. אם הוא קטן מהמתח הנמדד, מעגל לוגי דואג להשארת ה-"1" ב-MSB. אם מתח היציאה גדול מהמתח הנמדד, מאופסת סיבית זאת. לאחר מכן, קובעים "1" בסיבית הבאה וחוזרים על התהליך. כך עד ה-LSB. המספר הבינארי שהתקבל הוא המספר המבוקש. בצורה כזו זמן ההמרה הוא אחיד ושווה למספר סיביות אוגר ההזזה, המגדיר את יכולת ההפרדה (הרזולוציה) של ה-ADC.

סוג אחר של רכיבי ADC הם ה-ADC המהירים, המכונים גם Flash ADC. זמן ההמרה שווה למהירות תגובה של מגברי שרת ומגיע עד כדי עשרות ns בלבד. רכיבים אלה כוללים שורה ארוכה של מגברי שרת, שכל אחד משווה את מתח הכניסה עם מתח ייחוס מסוים. לדוגמא, ADC ברזולוציה של 8 סיביות יכיל 256 מגברי שרת, הכוללות את כל אפשרויות ההמרה.

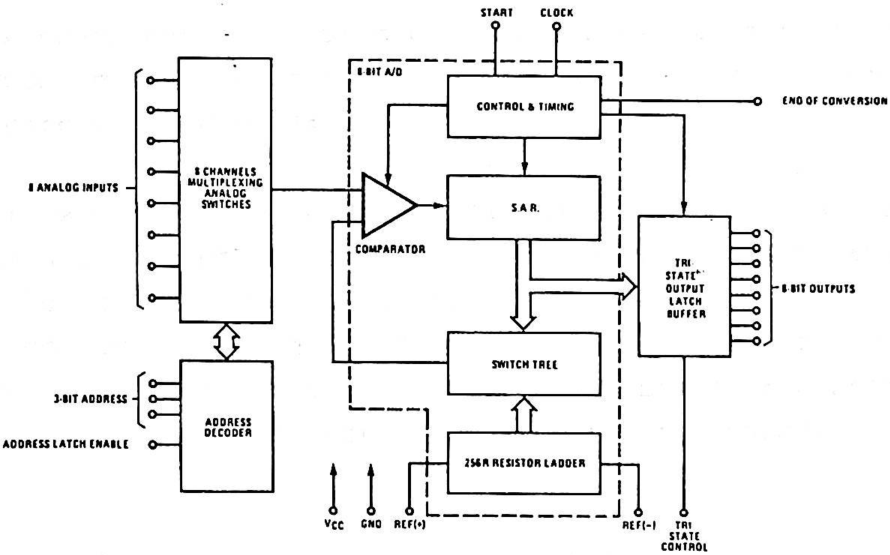
7.4 <u>ADC מונוליתי בריבוב ערוצים - ADC 0808</u>

אין לסיים דיון ב-ADC מבלי לתאר את רכיבי ה-ADC המונוליתיים המותאמים למיקרו-מחשב. רכיבים אלה כוללים בתוכם את כל מעגל ההמרה, המתואר בסעיף הקודם בשיטות המרה שונות, כולל חוצץ ביציאה. כל שהם זקוקים לשם

הפעלתם הוא פולסי שעון בתדר מתאים ופולס START להתחלת ההמרה. בסיום המרה הם מוציאים את EOC. פניה לערוץ היציאה גורמת להוצאת הנתון לקוי המוצא.

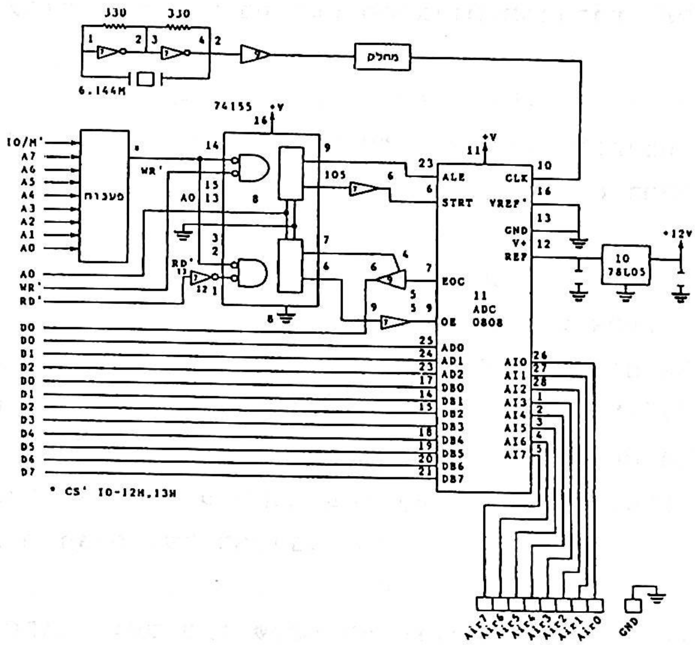
היום מקובל להשתמש ברכיבים מתוחכמים יותר מ-ADC למתח כניסה בודד. משתמשים ברכיב המרבב (Multiplexer) מספר כניסות. נתאר רכיב המרבב 8 כניסות אנלוגיות. לרכיב זה יש, מלבד הקוים הרגילים של ADC, 3 כניסות כתובת. ניתן לחבר אליו 8 כניסות אנלוגיות שונות. פניה לרכיב עם כתובת מתאימה ב-3 קוי כתובת אלה, ממירה מתקבילי (analog) לספרתי (digital) את הערוץ שזוהי כתובתו.

רכיב כזה הוא ה-ADC0808 והוא מתואר בציור 7-7.



ציור 7-7 תיאור ה-ADC0808

הציור הבא מתאר חיבור ה-ADC 0808ל ל-µP להמרת אותות תקביליים בתחום 0-5V.



ציור 8-7 אופן חיבור ADC0808 ל-BUS המיקרו-מחשב

כדי להמיר את תקבילי מאחד משמונת ערוצי הכניסה של ה-ADC יש לבצע ארבע פניות אליו. פניה אחת לקביעת הערוץ הנדגם. פניה שניה למתן אות SOC (Start Of Conversion). פניה שלישית לדגימת קו EOC (End Of Conversion) עד לזיהוי סוף ההמרה. פניה רביעית לאיסוף המספר המתקבל מההמרה.

המעגל המתואר תופס ארבע כתובות I/O. שתיים כפורט מוצא ושתיים כפורט מבוא. לשם כך משתמשים במעגל המפענח את הקוים 'IO/M וקוי הכתובת A1-A7. את יציאת מעגל הפענוח, קו A0 וקוים 'RD ו-'WR מחברים ל-74LS155. רכיב זה כולל שני מפענחים 1 מתוך 4. האחד יוצר קו 'CS כאשר ה-CPU כותב והאחר כאשר ה-CPU קורא. הכתובות אותן תופס המעגל הן IO-12H ו-IO-13H.

בכתיבה לכתובת IO-12 מופעל קו Address Latch) ALE Enable). זה נועל בתוך ה-DAC את מספר הערוץ להמרה. מספר הערוץ צריך להופיע בשלושת הסיביות הנמוכות של המספר הנכתב.

כתיבה של מספר כלשהו (לא חשוב איזה) לכתובת IO-13, גורמת ליצירת דופק SOC, הגורם להתחלת ההמרה ברכיב.

קריאה מכתובת IO-12, קוראת לסיבית D0 את קו EOC של הרכיב. כאן זקוקים לפורט מבוא בן סיבית אחת. לשם כך משתמשים במעגל ברכיב 74LS125 הכולל 4 חוצצים שלכל אחד קו איפשור נפרד. ברגע שמזהים שקו EOC עלה ל-"1" קוראים את המספר המתקבל מההמרה מכתובת IO-13. קריאה מכתובת זו יוצר אות CS המגיע לקו OE (Output Enable), הפועל בלוגיקה חיובית, של הרכיב.

ה-ADC-0808 פועל לפי שיטת ה-Successive Approximation וזקוק לשם כך לגל ריבועי, שתדירותו קובעת את קצב ההמרה. אות זה מסופק לרגל 10 של הרכיב. לרגל 12 של הרכיב יש לספק מתח 5V מיוצב המשמש כמתח ייחוס (VREF+).

לעתים משתמשים ב-ADC לבקרה על חיישנים הרחוקים ממנו. חיישנים המגיבים לטמפרטורה, לחץ, לחות, אור וכדומה. חיישנים המקובלים בתחום זה ממירים את השינויים בפרמטר המבוקש לשינויי זרם ולא לשינויי מתח. זאת כדי שמפלי המתח בדרך לא ישפיעו. החיישנים ניזונים ממקור מתח ישר כלשהו. בדרך כלל תחום הזרמים המקובל בתעשיה הוא התחום 4-20mA ולכן השיטה נקראת current loop 4-20mA. כדי להמיר זרם זה למתח, ניתן לחבר נגד בין כניסת ה-ADC לאדמה. הזרם שיגיע לנגד זה מהחיישן יומר על-ידי הנגד למתח. אם נחבר למשל נגד של 250Ω לכניסה, נקבל בזרם של 20mA מתח של 5V (הערך המכסימלי לכניסת ה-ADC המתואר).

מצפים מחיישן שיתנהג בצורה ליניארית. חיישנים כאלה יקרים בדרך כלל. כאשר מחברים חיישן ל-ADC המחובר ל-µP, ניתן להשתמש גם בחיישן לא ליניארי. במקרה כזה, מוסיפים לתוכנת הדגימה תוכנית, המבוססת על טבלת חיפוש, הממירה את הקריאות הלא ליניאריות לקריאות ליניאריות.

7.5 שאלות ותרגילים

7.5.1 שרטט מעגל חשמלי מפורט של DAC בהתאם לציור 7-1. למעגל המוצא חבר מגבר מהפך בעל הגבר של 1/3- והתנגדות כניסה של 2KΩ. השתמש במגבר שרת 741 כאשר R הוא בעל ערך של 2KΩ.

7.5.2 שרטט מעגל חשמלי מפורט של DAC בהתאם לציור 7-2. למעגל המוצא חבר מגבר בעל הגבר של 1/4- והתנגדות כניסה של 5KΩ. השתמש במגבר שרת 741 כאשר R הוא בעל ערך של 5KΩ.

7.5.3 שרטט מעגל חשמלי מפורט של ADC בהתאם לציור 7-5.
כ-DAC השתמש במעגל של ציור 7-3.
כמונה 8 סיביות, השתמש ברכיב 4040.
כרכיב בעל 8 תאי DFF, השתמש ברכיב 74LS374.
בנה את המתנד בעזרת 555.

7.5.4 שרטט מעגל חשמלי מפורט של ADC בהתאם לציור 7-6.
כ-DAC השתמש במעגל של ציור 7-3.

7.5.5 שרטט מעגל חשמלי מפורט כולל מעגל פענוח לרכיב ADC0808. חבר את יציאותיו ישירות למסלול הנתונים. את יציאת EOC חבר לסיבית D0 של פורט IN-1.

7.5.6 כתוב תוכנית הפונה לכתובת ערוץ 3 ב-ADC שחיברת בסעיף הקודם לשם התחלת ההמרה, לאחר מכן בודקת בלולאה אינסופית את סיבית D0 בפורט IN-12. ברגע שהיא מבחינה בעליית קו זה ל-HIGH, היא קוראת

מכתובת ערוץ 3 את הנתון ומוציאה אותו דרך פורט
מוצא B לנוריות. בצע תוכנית זו בלולאה אינסופית.

תרגול

7.6 הפעלת DAC מונוליתי

במעגל DSM-2200 נמצא DAC הנקרא DAC-08. לרכיב 8 כניסות ספרתיות המגיעות מפורט B של ה-8255. ביציאת הרכיב מופיע זרם, שהוא פונקציה של המספר הבינארי שבכניסותיו. זרם זה מסופק למגבר שרת שממירו למתח בתחום 0-5V. המעגל מתואר בציור 7-3.

המעגל כולל טרימר פוטנציומטר, המשמש לכיול חד פעמי של מגבר השרת. כאשר מקבל ה-DAC 00 בכניסותיו, מוציא מגבר השרת 0V. כאשר מקבל ה-DAC FF בכניסותיו, מוציא מגבר השרת 5V.

7.6.1 חבר לכניסות ה-8255 את מעגל נוריות/מפסקים DSM-2150.

7.6.2 רשום תוכנית המבצעת היתחול ל-8255 כך שפורט A ופורט B הם פורטי מוצא ופורט C הוא פורט מבוא. לאחר מכן קוראת התוכנית נתון מפורט C ומוציאה אותו לפורטים A ו-B בלולאה אינסופית.

7.6.3 הרץ את התוכנית.

7.6.4 שנה את המפסקים ומדוד את המתח, המופיע ביציאת מעגל ה-DAC (ANO) וה-GND.

7.6.5 הפסק את התוכנית.

7.6.6 רשום תוכנית המבצעת היתחול ל-8255 כמו בסעיף 7.6.2, מוציאה ל-DAC מספרים גדלים בהדרגה, עד ל-FF ולאחר מכן מספרים קטנים בהדרגה על ל-00 וחוזר חלילה. בין הוצאת מספר אחד לשני מבצעת השהייה לפי מצב המפסקים בפורט C.

7.6.7 הרץ את התוכנית.

7.6.8 חבר סקופ ליציאת מעגל ה-DAC. צייר את הגל המתקבל. שנה את המפסקים, ובדוק את ההשפעה על צורת הגל.
בדוק את הליניאריות של הגל ואת גודל המדרגה.

7.6.9 הצע דרך ליצירת גל סינוסואידלי.

7.7 מימוש ADC בעזרת DAC

בערכה S-2100 נמצא מגבר שרת משווה חופשי, שכניסותיו ויציאתו מחוברים ללוח תקיעה קטן ליד הבחון הלוגי. נבנה מעגל כמתואר בציור 6-7.

7.7.1 חבר את יציאת מעגל ה-DAC לכניסה הלא המהפכת (כניסת +) של מגבר השרת.

7.7.2 חבר לכניסה המהפכת (כניסת -) של מגבר השרת יציאת זחלן של פוטנציומטר ששני הדקיו האחרים מחוברים ל-+5V ול-GND. הפוטנציומטר יספק את המתח הנמדד.

7.7.3 חבר את יציאת מגבר השרת לכניסת D0 של פורט מבוא A.

7.7.4 שנה את הפוטנציומטר כך שהמתח הנמדד יעמוד על 2.5V.

7.7.5 רשום תוכנית המבצעת:

א) היתחול ל-8255 כמתואר בסעיף 7.6.2.

ב) מוציאה 00 לפורט B.

ג) בודקת סיבית של פורט מבוא A.

ד) אם סיבית זו ב-"1", סימן שהמתח הנמדד קטן ממתח ה-DAC. דלג לסעיף ז'.

ה) הגדל המספר בפורט B ב-1.

ו) דלג לסעיף ג'.

ז) הוצא את המספר שהוצאת לפורט B לפורט A (הנוריות) של ה-8255.

ח) דלג לסעיף ב'.

7.7.6 הרץ את התוכנית.
בנוריות של פורט A של ה-8255, צריך להתקבל 80H בערך.

7.7.7 שנה את הפוטנציומטר.
המספר בנוריות צריך להשתנות בהתאם.

שים לב, שגם כאשר הפוטנציומטר קבוע, המספר בדרך כלל משתנה מעט. הדבר נובע מערכים גבוליים ושינויים במתח הספק.

7.8 שימוש ב-ADC-0808

בכרטיס DSM-2200 נמצא ADC-0808, כמתואר בציור 7-8. אופן הפעלתו מתואר בסעיף 7.4 בפרק.

7.8.1 בדוק חיבור כרטיס DSM-2200 לערכה S-2100. חבר את כרטיס נוריות/מפסקים DSM-2150 ל-8255.

7.8.2 חבר פוטנציומטר בין +5V ו-GND. את הזחלן שלו חבר לכניסת ANIO של ה-ADC-0808.

7.8.3 רשום תוכנית המבצעת:
א) היתחול ל-8255.
ב) המרת המתח מכניסת ערוץ ANIO של ה-ADC-0808 לבינארי בהתאם לאלגוריתם המתואר בסעיף 7.4.
ג) מוציאה את הערך הבינארי לנוריות המחוברות לפורט A של ה-8255.
ד) מדלגת לסעיף ב'.

7.8.4 הרץ את התוכנית.

7.8.5 שנה את הפוטנציומטר. מדוד את המתח המסופק לכניסת ה-ADC והשווה את המתח הנמדד עם המספר הבינארי המתקבל.

7.9 דגימת אותות באמצעות ADC ושיחזורם באמצעות DAC

מטרת התרגיל היא לדגום אותות בעזרת ADC ולאגור את הערך הבינארי של הדגימות בזכרון. לאחר מכן לשחזר את האותות על-ידי העברת הערכים הבינאריים של הדגימות למעגל DAC.

7.9.1 חבר למבוא ה-ADC גל סינוס בתדירות של 0.1HZ כאשר VMAX=5V ו-VMIN=0V.

7.9.2 כתוב תוכנית כמו בסעיף 7.8.3, הדוגמת את הכניסה ומוציאה לפלט. יחד עם ההוצאה לפלט, מוציאה התוכנית את המספר לטבלה. בין דגימה לדגימה בצע השהייה בתוכנית של 0.5 שניות.

7.9.3 הרץ תוכנית זו למשך 10 שניות והתבונן בפלט.

7.9.4 העתק למחברתך את המספרים שקבלת בטבלה. המר אותם לעשרוני. שרטט גרף על נייר מילימטרי.

7.9.5 כתוב תוכנית המוציאה את המספרים האגורים בטבלה, בזה אחר זה, ל-DAC בלולאה אינסופית. בין הוצאת כל מספר מתבצעת השהייה בתוכנית של 0.5 שניות.

7.9.6 חבר את כניסת המשקף למוצא ה-DAC.

7.9.7 הרץ את התוכנית והתבונן בצורה המתקבלת במסך. שרטט צורה זו.

7.9.8 שנה את ההשהייה בין הוצאת המספרים להשהייה של 0.05 שניות. חזור על סעיף 7.9.7. לאיזה מסקנה אתה מגיע מתוצאות תרגיל זה.

פרק 8

תקשורת בין מחשבים

8.1 סיווג שיטות תקשורת

בתקשורת בין מחשבים, קשורים המחשבים ביניהם דרך קוי תקשורת. בכל שלב בתקשורת קיים מחשב משדר ומחשב מקבל. המשדר שולח את האינפורמציה שלו דרך פורט מוצא והמקבל מקבל אינפורמציה זו דרך פורט מבוא. יתכן מצב שהמחשב המשדר ישדר ולאחר מכן יעבור למצב קליטה ולהיפך, אבל, לא יתכן שיהיה בשני המצבים בו זמנית. כל מחשב אינו רואה את הנעשה אצל משנהו. הם מבחינים רק באינפורמציה הנמצאת בפורט הכניסה שלהם, לכן חלק מהאינפורמציה המועברת הוא הודעות הקשורות למצבי המחשבים המשדרים והמקבלים. הודעות כמו: "מוכן לקבלת הודעה", "קבל הודעה", "סוף הודעה" וכו'. יתכן מצב, עקב תקלות בערוץ התקשורת, שהמידע הנמסר הוא לא המידע שמגיע.

<u>שיטות התקשורת למיניהם מסווגות בשלוש צורות מיון:</u>

<u>א. סינכרונית ואסינכרונית:</u>

בתקשורת סינכרונית, קשורים המחשבים לקו משותף המספק אותות סינכרון לשניהם. אותות הסינכרון מאפשרים למחשבים לדעת מתי לשדר ומתי לצפות להודעה בקוי התקשורת. כל מחשב, לפני שהוא משדר הודעה, ממתין להיווצרות אות הסינכרון ואז שולח אותה. מחשב, שאמור לקבל את ההודעה, ממתין להופעת אות הסינכרון ואז אוסף אליו את המידע הנמצא בקוי פורט המבוא שלו.

בתקשורת אסינכרונית חוסכים את הקו של פולסי הסינכרון ומחולל הפולסים. על קוי האינפורמציה שולחים בתחילת כל הודעה אות התחלה. המחשב המקבל ממתין לקבלת אות התחלה זה. כאשר הוא מוצא אות זה הוא אוסף את ההודעה שבהמשך. תקשורת זו מקובלת יותר.

ב. מקבילית וטורית:

בתקשורת מקבילית שולחים את המידע במקביל. נתון בן 8 סיביות נשלח דרך כבל בן 8 גידים כאשר כל סיבית נשלחת על גיד נפרד בו זמנית. שיטה זו מחייבת כבל בעל מספר גידים רב.

בתקשורת הטורית משתמשים במספר גידים קטן. את הנתון שולחים דרך קו אחד, סיבית אחר סיבית. המשדר והמקבל חייבים להיות מתואמים מבחינת קצב השידור והקליטה.

ג. Polling או פסיקה:

הבעיה בתקשורת היא לדעת מתי מתחיל הדו-שיח. אחת השיטות היא לקבוע את אחד המחשבים כשולט (MASTER) והאחרים כשלוחות (SLAVES). השולט תמיד יוזם את התקשורת. הוא פונה אל השלוחה ושואל אם יש לה מה למסור, וממתין זמן מסוים לקבלת הודעה מהשלוחה. אם לא קבל, חוזר לתוכנית הראשית שלו. הפניה שלו אל השלוחה נעשית כל מירווח זמן מוגדר. כאשר לשלוחה יש מה למסור, היא ממתינה לפניה של השולט אליה, ושולחת לו מידע. השולט מגיב ואוסף את המידע. שיטה זו נקראת שיטת תקשורת על-ידי Polling.

שיטה אחרת להתחלת שיחה היא בעזרת פסיקה. משתמשים בפורטי מבוא בעלי קו STB (STroBe). כאשר מחשב אחד רוצה לפנות למחשב אחר, הוא שולח לפורט הכניסה שלו את ההודעה יחד עם דופק Strobe. פורט המבוא אוסף אליו את הנתון ומבצע

פסיקה למחשב המקבל. האחרון עובר לביצוע תוכנית הפסיקה, המטפלת בנתון הנמסר. שיטה זו מהירה ונוחה אם כי מחייבת שימוש בפורטים מתאימים ובתוכניות פסיקה.

מושג נוסף שקיים בתקשורת הוא המושג Handshake (לחיצת יד). במושג זה מתכוונים ששולח הודעה ממתין לאישור ממקבל ההודעה על קבלתה. ללא אישור זה אין הוא ממשיך בתוכנית.

נתאר חלק מהשיטות.

8.2 <u>תקשורת מקבילית אסינכרונית</u>

תקשורת מקבילית אסינכרונית, הנפוצה ביותר בעולם, היא התקשורת בין מחשב למדפסת. נוהל תקשורת זה נקרא גם פרוטוקול Centronics (אחת מיצרניות המדפסות הגדולות בעולם). נתאר נוהל תקשורת זה.

כדי להעביר מידע בצורה אמינה ואחידה בין מחשבים, קבעו צופן סטנדרטי לשם כך. לכל תו (ספרה, אות או סימן כתב) קבעו ערך בינארי. ערך בינארי זה משודר ממחשב למחשב בצופן הנקרא צופן ASCII שפירושו American Standard Code for Information Interchange - צופן אמריקאי סטנדרטי להחלפת אינפורמציה. הטבלה הבאה היא טבלת הצפנים. המספרים הבינאריים בתחום 0-1F משמשים כתווי בקרה. למשל, המספר 0A משמש כהוראה למדפסת "קדם שורה". או המספר 0D משמש כהוראה Carriage Return - נוע לתחילת השורה.

טבלת צפנים

00 - NUL	10 - DLE	20 - SP	30 - 0	40 - @	50 - P	60 - ’	70 - p
01 - SOH	11 - DC1	21 - !	31 - 1	41 - A	51 - Q	61 - a	71 - q
02 - STX	12 - DC2	22 - "	32 - 2	42 - B	52 - R	62 - b	72 - r
03 - ETX	13 - DC3	23 - #	33 - 3	43 - C	53 - S	63 - c	73 - s
04 - EOT	14 - DC4	24 - $	34 - 4	44 - D	54 - T	64 - d	74 - t
05 - END	15 - NAK	25 - %	35 - 5	45 - E	55 - U	65 - e	75 - u
06 - ACK	16 - SYN	26 - &	36 - 6	46 - F	56 - V	66 - f	76 - v
07 - BEL	17 - ETB	27 - '	37 - 7	47 - G	57 - W	67 - g	77 - w
08 - BS	18 - CAN	28 - (	38 - 8	48 - H	58 - X	68 - h	78 - x
09 - HT	19 - EM	29 -)	39 - 9	49 - I	59 - Y	69 - i	79 - y
0A - LF	1A - SUB	2A - *	3A - :	4A - J	5A - Z	6A - j	7A - z
0B - VT	1B - ESC	2B - +	3B - ;	4B - K	5B - [	6B - k	7B - {
0C - FF	1C - FS	2C - ,	3C - <	4C - L	5C - \	6C - l	7C - |
0D - CR	1D - GS	2D - -	3D - =	4D - M	5D -]	6D - m	7D - }
0E - SO	1E - RS	2E - .	3E - >	4E - N	5E - ^	6E - n	7E - ~
0F - SI	1F - US	2F - /	3F - ?	4F - O	5F - _	6F - o	7F - DEL

צופן ASCII הוא צופן בן שבע סיביות, כלומר בעל 128 צירופים. ב-128 הצירופים הבאים (מ-80 עד FF), משתמשים במחשבים לתוים מיוחדים ולסימנים גרפיים בדרך כלל.

כאשר קובעים נוהל תקשורת בין מחשבים קובעים גם את צורת החיבור. משתמשים במחברים סטנדרטיים וקובעים תפקיד לכל רגל. נתייחס לחלק מהקוים של התקשורת וניצור תקשורת אל מדפסת.

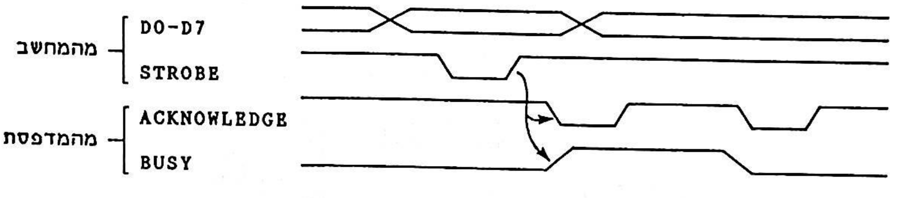
אם נתבונן במחבר הנמצא בקצהו של כבל מדפסת (בצד של המדפסת) נראה שלכל רגל יש מספר (או לפחות מסומנת רגל 1). את המספר הבינארי המציין את התו המשודר שולחים דרך הרגליים 2 עד 9, כאשר D0 היא רגל 2 ו-D7 היא רגל 9. רגל 1 משמשת כקו תזמון (Strobe). דרך קו זה מודיעים למדפסת שמידע חדש ממתין בקוים 2 עד 9. הודעה זו נעשית על-ידי הורדת הקו ל-"0" והעלאתו ל-"1". את הקוים 1 עד 9 מחברים לקוים של פורטי מוצא.

לא תמיד פנויה המדפסת לקליטת תוים. בדרך כלל, קולטת המדפסת תוים בזה אחר זה לתוך זכרון פנימי שלה (הנקרא buffer). כאשר היא קולטת את התו 0A, שמשמעותו "קדם שורה", היא מדפיסה את כל השורה. ישנן מדפסות שה-buffer שלהן גדול יותר ומגיע עד אלפי תוים.

המדפסת מסמנת למחשב את מצבה בעזרת קוים נוספים. קו אחד הוא קו BUSY (רגל 11). בעזרת קו זה מודיעה המדפסת באם היא מוכנה לקבל נתון (BUSY="0") או שהיא עסוקה (BUSY="1"). המחשב, לפני שהוא שולח תו למדפסת, דוגם קו זה. אם הוא ב-"1", הוא ממתין. כאשר הקו ב-"0" הוא שולח את המידע. קו זה מחברים לפורט מבוא במחשב.

קו נוסף למידע מהמדפסת הוא קו Acknowledge (רגל 10). בעזרת קו זה מאשרת המדפסת את קליטת המידע. קו זה יורד ל-"0" זמן קצר לאחר קבלת אות ה-Strobe.

לסיכום, דיאגרמת הזמנים של התקשורת עם המדפסת תיראה בצורה הבאה:



אלגוריתם התקשורת יהיה כדלקמן:

א. בדיקה, האם קו BUSY (רגל 11) נמצא ב-"0". אם לא, המתנה עד שירד ל-"0".
ב. מסירה לקוי ה-DATA (רגליים 2 עד 9) את צופן התו המיועד להדפסה.
ג. הורדת קו Strobe (רגל 1) ל-"0" והעלאתה ל-"1".

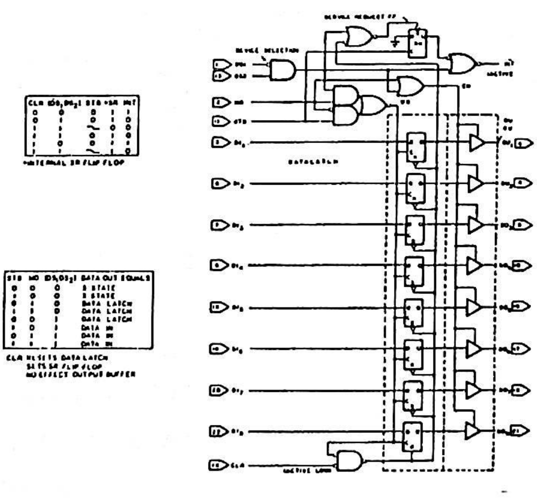
ד. בדיקת קו Acknowledge (רגל 10), אם ירד ל-"0" ועלה ל-"1". ניתן לוותר על בדיקה זו.

מלבד קוים אלה קיימים קוים נוספים כמו קו המציין אם נמצא נייר במדפסת (רגל 12) וקו המציין אם המדפסת במצב "On Line" (רגל 13). ניתן לחבר גם קוים אלה לפורט מבוא ולהתייחס אליהם בתוכנה או להתעלם מהם.

8.3 תקשורת מקבילית בעזרת פסיקה

בשיטה המתוארת בסעיף הקודם, עסוק המחשב המשדר במשך כל זמן ההדפסה. המדפסת איטית בהרבה מהמחשב. כדי להתגבר על בעיה זו, משתמשים ברוטינות פסיקה. מחברים את כניסות המידע של המדפסת לפורט מוצא במחשב המשדר, הכולל קו STB וקו פסיקה. ברגע שקו STB של פורט המוצא יורד ל-"0", נוצרת בקשת פסיקה למחשב המשדר, והוא משדר תו למדפסת.

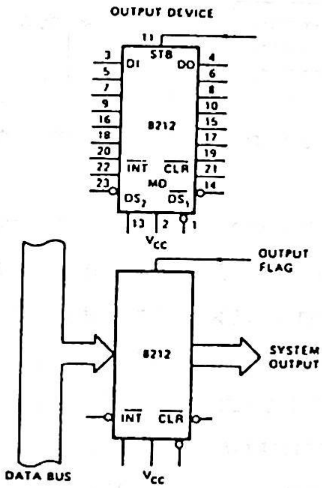
נדגים זאת בעזרת הרכיב 8212 ולאחר מכן בעזרת ה-8255.



ציור 8-1 תיאור ה-8212

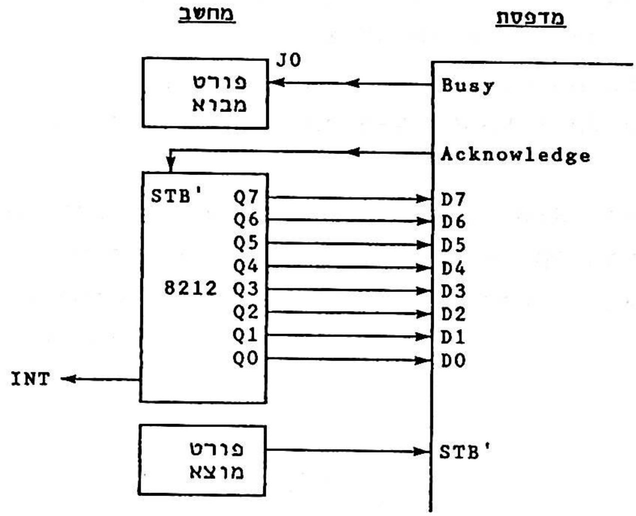
ה-8212 כולל תאי Latch שיציאותיהם נשלטות על-ידי חוצצים. לרכיב 2 קוי 'CS (DS0-ו DS1'). מלבד מרכיבים אלה המאפיינים פורטים, יש לו קוים נוספים. קו 'CLR הגורם לאיפוס תוכן תאי ה-Latch. קו MD הקובע אם הוא עובד ב-MODE OUT ("MD="1) או ב-MODE IN ("MD="0).

הקוים המאפשרים ביצוע פסיקה הם הקוים 'STB ו-'INT. את קו INT מחברים לקו פסיקה של ה-CPU. אם קו הפסיקה מופעל ב-High, מחברים אותו דרך מהפך. הורדת קו 'STB מ-High ל-Low גורמת לבקשת פסיקה בקו 'INT.



ציור 2-8 חיבור ה-8212 כפורט מוצא

נחבר את ה-8212 למדפסת בצורה הבאה:



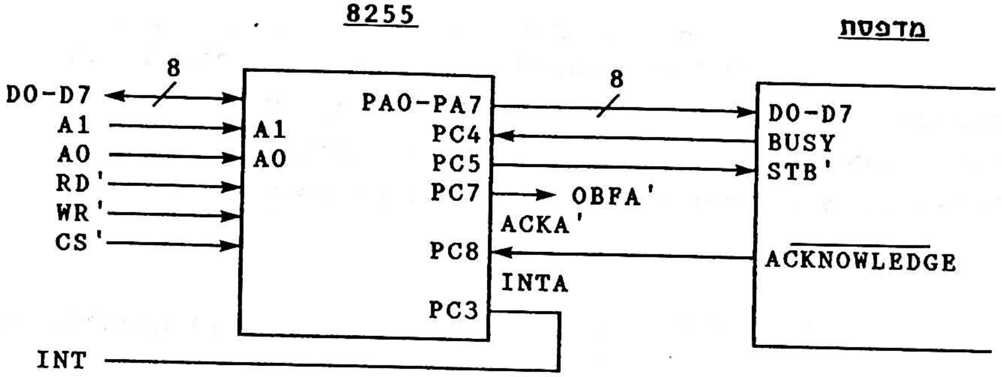
ציור 3-8 חיבור למדפסת בעזרת 8212

ברגע שהתוכנית הראשית רוצה לשלוח בלוק נתונים להדפסה במדפסת, היא טוענת אותו ל-RAM ורושמת בסוף הבלוק נתון מסוים כזקיף (נניח FF לשם דוגמא). היא רושמת את כתובת התחלת הבלוק בזוג תאים המשמשים כוקטור המציין את כתובת הנתון המיועד לשידור. לאחר מכן, היא מאפשרת פסיקה ומשדרת את הנתון הראשון. לאחר שידור הנתון הראשון היא מסיימת את חלקה בשידור הבלוק.

שידור הנתון הראשון גורם להיווצרות Acknowledge מהמדפסת וזה גורם לבקשת פסיקה. רוטינת הפסיקה בודקת אם המדפסת במצב BUSY. אם כן, חוזרת לתוכנית הראשית ואם לא, פונה לתא, שהוקטור מצביע עליו. אם הנתון איננו הזקיף היא משדרת אותו למדפסת, מקדמת את המצביע וחוזרת לתוכנית הראשית. אם הנתון הוא הזקיף, חוסמת את הפסיקה וחוזרת לתוכנית הראשית. שידור תו נעשה על-ידי כתיבתו ל-8212 והורדת קו STB של המדפסת ל-"0" והעלאתו ל-"1" בעזרת פורט המוצא.

אם המדפסת היתה במצב BUSY, היא מודיעה על יציאה ממצב זה בעזרת Acknowledge נוסף. זה יגרום לבקשת פסיקה ולשידור הכתון הבא.

ניתן להשתמש ב-8255 במקום המערכת שתוארה בציור הקודם.



ציור 4-8 חיבור למדפסת בעזרת 8255

אלגוריתם התקשורת זהה לאלגוריתם ה-8212. עם כתיבת המידע על-ידי ה-CPU בפורט A, יורד קו OBFA ל-"0". הוא היה יכול לשמש כקו STB למדפסת אלמלא העובדה, שנדרשת השהייה מסוימת בקו זה, על מנת לאפשר התייצבות הנתונים לפני מתן האות. אם רוצים לחבהו ולחסוך את הורדת קו PC5 ל-"0" והעלאתו ל-"1", מחברים חד יציב בינו לבין קו STB של המדפסת לשם השהייה.

8.4 תקשורת טורית אסינכרונית

תקשורת טורית היא תקשורת בה מתרגמים את הנתון בן 8 הסיביות לסידרה של דפקים טוריים, אפס ואחד, המשודרים דרך קו תקשורת אחד. המקבל יודע את משך הזמן של כל דופק

שמשדר המשדר.

בתקשורת טורית אסינכרונית קיימת בעיה של זיהוי ההתחלה של כל נתון. לכן קבעו את הנוהל הבא:

<u>סיבית התחלה - Start bit:</u>

הקו במצבו הנורמלי נמצא ב-HIGH. לפני כל נתון, הנשלח בצורה טורית, חייבת לבוא סיבית אפס למשך זמן המוקצב להעברת כל סיבית. סיבית זו נקראת סיבית התחלה. מקבל הנתונים מזהה התחלת שידור תו על-ידי זיהוי המעבר מ-"1" ל-"0".

<u>סיביות הנתון - Data bits:</u>

עם סיום השידור של סיבית ההתחלה, משודרות 7 סיביות הנתון בזו אחר זו. כל אחת במשך זמן מוגדר וקבוע. מכיוון שמקבל הנתונים יודע מתי התחיל השידור, הוא מבצע את דגימת סיביות הנתון באמצע של כל סיבית. זאת כדי להתגבר על הבעיה של תופעות מעבר.

<u>סיבית ביקורת - Parity bit:</u>

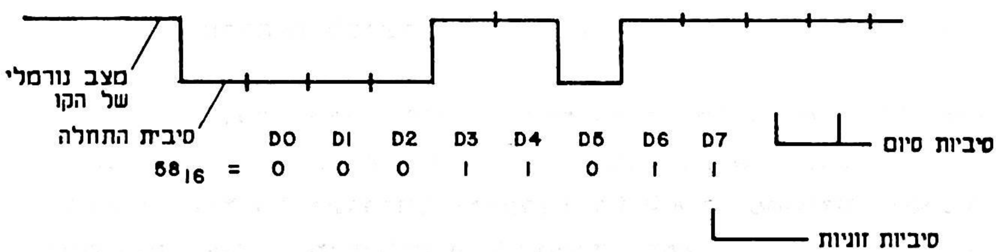
בדרך כלל, ממירים כל נתון לצופן ASCII ואותו משדרים. צופן ASCII הוא צופן בעל שבע סיביות. בסיבית השמינית משתמשים לבדיקת זוגיות. סיבית זו משמשת את המקבל לבדיקת הנכונות של הנתון שקיבל. ערך הסיבית ("0" או "1") נקבע לפי מספר ה-"1" שבנתון. קיימות שתי שיטות. ביקורת זוגיות וביקורת פרט. בביקורת זוגיות, מספר ה-"1", כולל סיבית הביקורת חייב להיות זוגי. לדוגמא, אם בנתון יש שלושה "1", קובע המשדר את סיבית הביקורת ל-"1". אם יש ארבעה "1", תהיה סיבית הביקורת "0".

בביקורת פרט, מספר ה-"1", כולל סיבית הביקורת, יהיה פרט.

<u>סיביות סיום - Stop bits:</u>

בסוף כל נתון, משודרות סיביות של "1" לוגי, בדרך כלל שתיים. סיביות אלה משמשות להעברת הקו למצבו הנורמלי, למשך זמן המאפשר למקבל לעבד עיבוד ראשוני את הנתון שאסף ולהסתנכרן מחדש על התחלת שידור התו הבא.

לסיכום, שידור תו בודד (58H) נראה בצורה הבאה:



ציור 5-8 שידור התו 58H בתקשורת טורית אסינכרונית

קצב שידור נמדד ביחידות של BAUD שהם סיביות משודרות לשניה. יחידת קצב שידור אחרת היא סיביות לשניה שבה הכוונה לסיביות הנתון המועברות בשניה. לדוגמא, אם משדרים בקצב של 10 תוים לשניה מקבלים 110 BAUD. כל תו דורש לשידורו 11 סיביות שידור (כולל סיבית ההתחלה וסיביות הסיום). קצב זה גם שווה ל-80 סיביות לשניה (סיביות נתון).

8.5 פרוטוקולי תקשורת

כאשר מאפיינים תקשורת בין מחשבים, יש לקבוע את נוהל (פרוטוקול) התקשורת ביניהם. נוהל התקשורת מגדיר את כל הפרמטרים הקשורים בתקשורת. נוהל תקשורת מוגדר ומאורגן, בדרך כלל, במספר רמות או שכבות. כל רמה מגדירה פרמטרים שונים של צורת התקשורת, החומרה והתוכנה. נתאר כמה מהן.

הרמה הראשונה נקראת רמה פיזיקלית (Physical Protocol), רמת הממשק. זו הרמה המתארת את אופן זרימת האותות (טורית-מקבילית, סינכרונית-אסינכרונית), רמות מתח או זרם של האותות, מהם קוי מידע ומהם קוי בקרה וכן הלאה. רמה זו מתייחסת לכיצד מועבר בית אחד של מידע.

המחשב הקולט אינו בהכרח דוגם את קו התקשורת כל הזמן וממתין למידע מהמחשב המשדר. יתכן שהוא נכנס לקליטה באמצע שידור נתונים. ייתכן גם שחלק מהמידע ישתבש. מסיבות אלה, מוסיפים לבתי המידע המשודרים מידע נוסף, המאפשר לקולט לזהות את תחילתה וסופה של האינפורמציה ולזהות שגיאות ושיבושים. המידע הנוסף מאפשר זיהוי ברמת בית בודד, למשל, סיבית התחלה, סיבית ביקורת וסיביות סיום.

ניתן להוסיף מידע לזיהוי ברמת בלוק, על-ידי הוספת תוים מתוך תוי הבקרה של צופן ASCII , המגדירים את תחילת הבלוק, סוף הבלוק, CHECKSUM וכד'. CHECKSUM הוא סכום בתי המידע המשודרים. המחשב הקולט מסכם את בתי המידע, המגיעים אליו, ומשווה אותם עם ה-CHECKSUM. אם אין זהות, הוא יודע שבבלוק שהגיע אליו ישנה טעות. באיפיון נוהל התקשורת מחליטים מה עושים במקרה של טעות. רמה זו של איפיון נוהל התקשורת נקראת רמת הערוק (Protocol Link).

כאשר למחשב המשדר מתוברים כמה מחשבים או מסופים, העשויים לקלוט את המידע, יש להוסיף למידע המשודר מידע נוסף. מידע זה מציין למי מיועדת האינפורמציה. אם המחשבים מאורגנים בארכיטקטורה פתוחה, שבה כל מחשב יכול לשדר לכל מחשב אחר, אזי יש לציין גם ממי היא שודרה. המידע המשודר מורכב ממנות. לכל מנה מוסיפים מידע, המתאר את השולח והנמען. רמה זו של נוהל התקשורת נקראת רמת הרשת (Network Protocol).

קיימות רמות נוספות בקביעת נוהל התקשורת כמו פרוטוקול ההובלה (Transport Protocol), פרוטורול המושב (Session Protocol), פרוטוקול ייצוג (Presentation Protocol), ופרוטוקול היישום (Application Protocol). אבל לא נתאר אותן בספר זה.

לא בהכרח זקוקים לכל הרמות. לדוגמא, בחיבור מסוף יחיד למחשב, זקוקים לרמה הפיזיקלית בלבד. מסוף הוא מכשיר, הכולל לוח מקשים וצג והקשור בתקשורת טורית למחשב. תוים המגיעים מהמחשב למסוף, נרשמים בצג שלו. לחיצה על לוח המקשים גורמת לשידור תו למחשב. אם מחובר מסוף אחד למחשב, אין צורך בהקמת קשר בכל רגע. המסוף דוגם כל הזמן את קו התקשורת ואת לוח המקשים. לכן התקשורת תהיה בתוים בודדים ולא בבלוקים של מידע. אם המסוף קרוב למחשב, גם אין צורך בבקרת שיבושים מסובכת, אם בכלל.

בעולם ישנם מכוני תקנים שונים, שקובעים נוהלי תקשורת. לדוגמא, EIA המהווה תקן התעשיה האלקטרונית האמריקאית (Electronic Industries Association) או ISO, הארגון הבינלאומי לתקנים (International Organization for Standartization).

נסביר עתה שלשה תקנים נפוצים לתקשורת טורית:

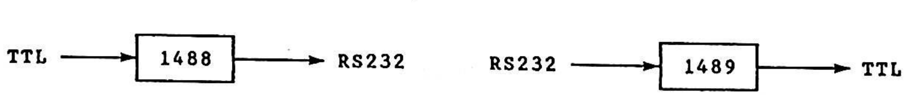
א. EIA-RS232.
ב. Current loop.
ג. EIA-RS422.

התקנים האלה שייכים לרמת הממשק - הרמה הפיזיקלית. בסעיף הקודם תוארה תקשורת טורית אסינכרונית. התקן קובע את רמת האותות, קוי הבקרה הנלווים, מחבר (Connector) התקשורת ותפקיד כל אחד מפיניו.

א. EIA-RS232:

בתקן זה, עובדים ערוצי התקשורת בלוגיקה שלילית. מצב "1" מצוין על קו התקשורת במתח בין 5V- ל-12V- (בדרך כלל 12V-). מצב "0" מצוין במתח בין 5V+ ל-12V+ (בדרך כלל 12V+). רכיבי התקשורת מוציאים, בדרך כלל, אותות בלוגיקה חיובית ברמת TTL (0-5V), ולכן זקוקים לדוחפים (דרייברים).

קיימים שני סוגי דוחפים. דוחף הממיר מרמת TTL לתקן RS232 ודוחף הממיר מתקן RS232 לרמת TTL. דוחפים פופולאריים כאלה הם הרכיבים 1488 ו-1489 המתוארים בציור הבא:



ציור 6-8 דוחפים ל-RS232

קיימים רכיבים הכוללים את שני סוגי הדוחפים במארז אחד, וישנם גם כאלה הניזונים מספק 5V+ יחיד וממירים את האותות לתקן RS232.

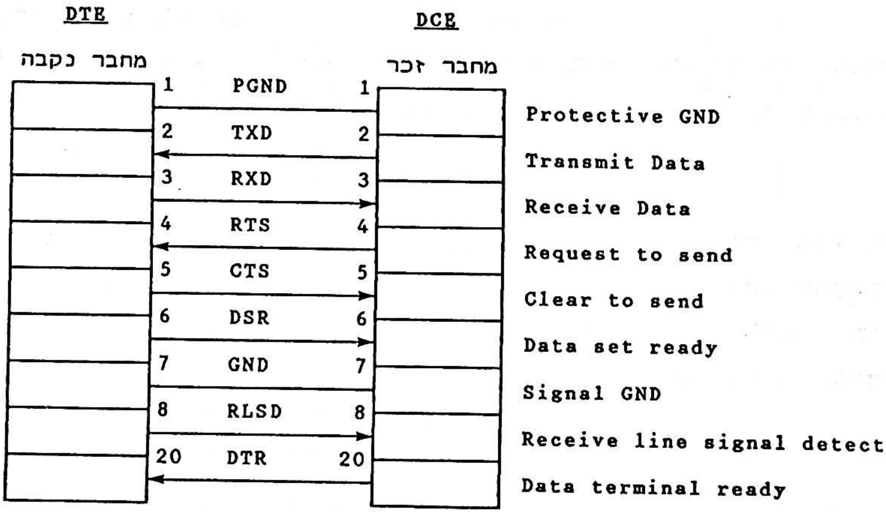
בתקן RS232 מחולקים כל סוגי הציוד לשתי קבוצות:

DTE - Data Terminal Equipment - הציוד הפסיבי יותר כמו מסופים, מדפסות וכד'.

DCE - Data Communication Equipment - הציוד האקטיבי יותר - המחשבים למיניהם.

התקשורת היא בדרך כלל תקשורת בין מחשב מקבוצת ה-DCE להתקן מקבוצת ה-DTE. גם כאשר מבצעים תקשורת בין שני מחשבים קובעים אחד כ-DCE ואת האחר כ-DTE. DCE מציין שהוא ה-Master במערכת התקשורת ו-DTE מציין שהוא ה-Slave. DTE משדר לפי רשותו של DCE.

כאמור, תקן תקשורת קובע גם את סוג המחבר ותפקוד כל פין בו. בתקן RS232 מקובל להשתמש במחבר מסוג D-Connector (בגלל צורתו זו ) בן 25 פינים. בצד ה-DCE משתמשים במחבר זכר ובצד ה-DTE במחבר נקבה. לא כל הפינים מנוצלים, ולכן ניתן למצוא גם מחברים אחרים. בנוסף לקו שידור וקו קליטה, מקשרים קווים נוספים בין המחשב להתקן. הקווים מתוארים בציור הבא.



ציור 7-8 תיאור מחבר RS232

שמות הקוים מתייחסים למעשה לצד ה-DCE. לדוגמא, ה-DCE משדר דרך רגל 2 (TXD), וה-DTE קולט דרך רגל זו. כדי להשתמש במונח קבוע לכל קו קבעו את השמות בהתאם ל-DCE.

PGND - (Protection GND) - זהו קו המשמש בדרך כלל לסיכוך של קוי התקשורת (מניעת רעשים). הוא מחובר לקו הארקה של המכשירים, במידה והם מוארקים.

TXD - (Transmit Data) - בקו זה משדר המחשב (DCE) להתקן (DTE).

RXD - (Receive Data) - בקו זה קולט המחשב DCE) מההתקן (DTE).

RTS - (Request To Send) - בקו זה מודיע המחשב להתקן על רצונו לשדר אליו אינפורמציה.

CTS - (Clear To Send) - בקו זה מודיע ההתקן למחשב, בתגובה לבקשת שידור שהגיעה אליו דרך קו RTS, על נכונותו או אי נכונותו לקלוט נתונים. המחשב אינו משדר עד שהוא אינו מזהה "1" בקו CTS.

GND - (GrouND) - זהו קו אדמה שרמות האותות מתייחסות אליו.

DSR - (Data Set Ready) - בקו זה מודיע ההתקן על רצונו לשדר מידע למחשב.

DTR - (Data Terminal Ready) - בקו זה מאשר המחשב להתקן, בתגובה לבקשת שידור שהגיעה אליו דרך קו DSR, על נכונותו לקלוט נתונים.

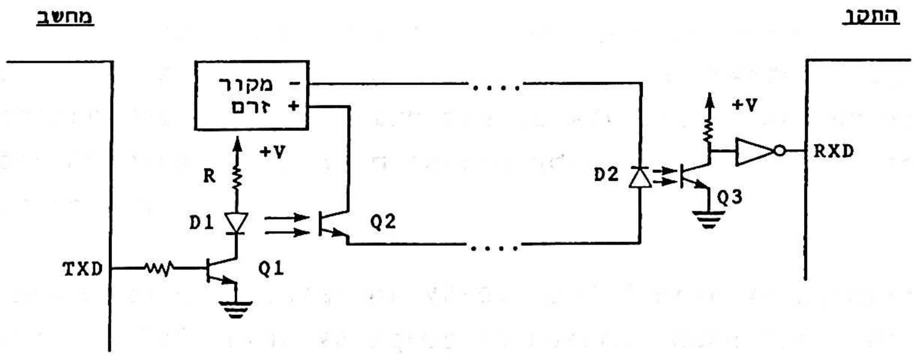
גם הקוים RTS, CTS, DSR ו-DTR פועלים ברמות מתח לפי תקן RS232 (±5-12V). לכן זקוקים גם עבורם לדוחפים. הרכיבים 1488 ו-1489 כוללים 4 דוחפים כל אחד. ניתן לחבר קוים אלה לפורטי מבוא ומוצא בצד המחשב, אם כי נכיר בהמשך רכיב המטפל בכל קוי התקשורת.

לעתים קרובות, יוצרים תקשורת טורית בעזרת שלושה קוים בלבד - TXD, RXD ו-GND. אם המחשב מתוכנן להתייחסות גם לקוי הבקרה, מקצרים את קו RTS ל-CTS ואת קו DSR ל-DTR. בצורה כזו מבקש השידור יוצר לעצמו את האישור לשידור.

ב. Current Loop:

מבחינה פונקציונלית של קוי התקשורת והבקרה, דומה תקן זה לתקן RS232. כל ההבדל הוא שרמות אותות התקשורת הן רמות זרם ולא רמות מתח. ב-"1" זורם זרם בקו וב-"0" אין זרם בקו. בדרך כלל, מדובר בזרם של 20mA, ולכן שיטה זו נקראת 20mA Current Loop.

בתקן Current Loop עבור כל אות קיים מקור זרם המחובר בחוג, בדרך כלל בצד המחשב. זה שמוסר את האות, סוגר או מנתק את החוג. הצד הקולט בודק את הקו ומזהה אם זורם בו זרם או לא. מיתוג וזיהוי הזרם מבוססים, בדרך כלל, על צמד אופטי, כמתואר בציור הבא:



ציור 8-8 תיאור תקשורת ב-Current Loop

ברגע ש-TXD עולה ל-High, ממותג טרנזיסטור Q1, זורם זרם דרך ה-LED המחובר אליו, ממתג את טרנזיסטור Q2 וסוגר אל החוג. הזרם שזורם דרך D2 גורם להארת D2, למיתוג Q3,

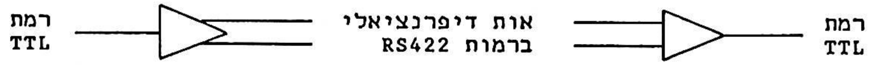
להופעת "0" בכניסה למהפך ו-"1" בכניסה RXD. פעולה הפוכה קורית במקרה ש-TXD יורד ל-Low.

מערכת זו מסורבלת מעט מתקן RS232. תקן זה מיועד למקומות בהם יש מפלי מתח גבוהים, הנובעים מהתנגדות גבוהה בקוי התמסורת. בחוג זרם, אין השפעה של ההתנגדות על הזרם, בהנחה שמקור הזרם מסוגל לעמוד בכך.

גם בתקן זה ניתן להשתמש בקוי הבקרה RTS, CTS, DSR ו-DTR.

<u>ג. תקן RS422:</u>

תקן זה דומה מאד לתקן RS232. ההבדל ביניהם הוא בדוחפים. כאן משתמשים בדוחפים דיפרנציאליים.



ציור 9-8 דוחפים ל-RS-422

משתמשים בתקן זה בסביבות רועשות מבחינה חשמלית, שבהן הסיכוך אינו מספיק. במגבר דיפרנציאלי הרעש מופיע על שני קוי התמסורת בצורה שווה ומונחת על-ידי המגבר. למגבר יחס דחיית אות משותף גבוה.

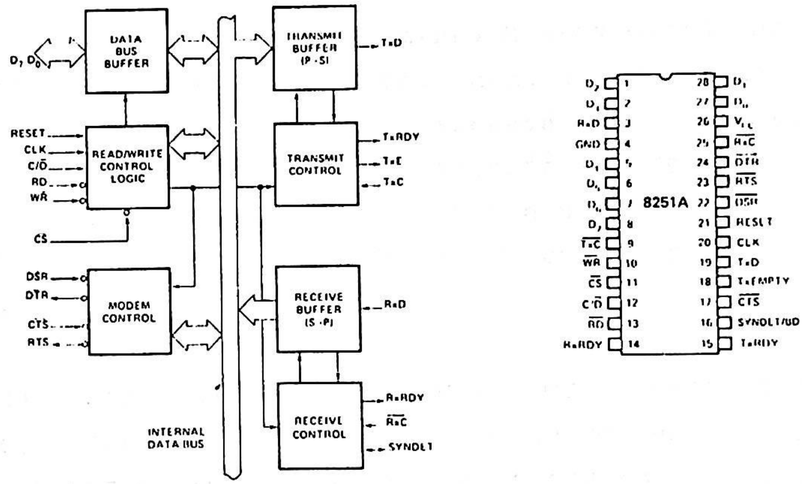
רמות המתח בתקן RS422 הן 0-5V. ב-"1" ניתן 5V בקוטביות אחת וב-"0" ניתן 5V בקוטביות הפוכה. האות "צף" ביחס לאדמה.

8.6 8251 USART

ה-USART הוא רכיב שבא לפשט מימוש תקשורת טורית מבחינת החומרה ובעיקר מבחינת התוכנה. משמעות המלה USART היא Universal Synchronous Asynchronous Receiver Transmitter - משדר מקלט כללי סינכרוני/אסינכרוני. זהו רכיב הניתן לתיכנות בהתאם לפונקציות שלו. ה-8251 הוא USART מהפופולאריים בעולם ובעל מספר רב מאד של פונקציות.

נתייחס לאופן העבודה האסינכרוני שלו. לאחר היתחול הרכיב, ידגום הרכיב בעצמו את קו התקשורת. אם יזהה מידע שמגיע בצורה טורית, יאסוף אותו אליו ועם גמר קליטת בית, יודיע על כך ל-CPU (בפסיקה או ב-Polling כפי שיתואר בהמשך). בצורה כזו, משוחרר ה-CPU מדגימת קו התקשורת והוא מקבל את המידע כאשר הוא מגיע, בצורה מקבילית. גם בשידור, מוסר ה-CPU את הבית המיועד לשידור ל-USART, כמו בכתיבה לפורט, וה-USART דואג להמרתו לצורה טורית בהתאם לקצב השידור המבוקש. ה- USART מסוגל לטפל בקוי הבקרה RTS, CTS, DTR, DSR וקוי בקרה נוספים.

בציור הבא מתוארת דיאגרמת המלבנים של ה-8251:



ציור 8-10   דיאגרמת מלבנים של ה-8251

ל-8251 שתי מערכות נפרדות לשידור ולקליטה. כלומר, הוא מסוגל לקלוט תו ולשדר תו בו זמנית, דבר המאפשר תקשורת Full Duplex בעזרתו. הרכיב הוא בן 28 רגליים.

8.6.1 תיאור הרגליים ותפקידיהם:

D0-D7 - קוי הנתונים. הרכיב מתחבר ישירות אל ה-CPU. ניתן לכתוב בו ולקרוא ממנו.

'CS - הרכיב תופס שתי כתובות בתחום I/O. כלומר, קו A0 לא משתתף במעגל הפענוח ליצירת 'CS.

'C/D - כתובת אחת של הרכיב מיועדת למילות בקרה (Command) והכתובת השניה לנתונים (Data). לקו זה מחברים את קו A0, המאפשר הבחנה בין הפניות השונות.

RESET - קו המחובר ל-RSTOUT של ה-CPU ומשמש להיתחול הרכיב.

'RD - מחובר לקו 'RD של ה-CPU לסימון קריאה מה-8251.

'WR - מחובר לקו 'WR של ה-בPU לסימון כתיבה ב-8251.
CLK - המערכת הלוגית ב-USART היא מערכת סינכרונית, המתוזמנת בעזרת אותות קו ה-CLK. לתדר ה-Clock יש ערכי מינימום ומקסימום, שקשורים לקצב השידור הנדרש ולאופן עבודה ה-8251. התחום הוא די רחב (740KHZ - 3.125MHz) ולא נדרש תדר מדויק.
DSR,DTR,CTS,RTS - קוי בקרת התקשורת הטורית, כמתואר בתקן RS232. לוגיקת קוים אלה מתהפכת כאשר מחברים אותם דרך דוחפים.

שאר הקוים מתחלקים לשתי קבוצות - אותות השידור ואותות הקליטה.

<u>אותות השידור:</u>

TXD (Transmit Data) - קו שידור המידע הטורי. הבית לשידור מגיע לחוצץ השידור במקביל והוא ממירו לסידרת סיביות המשודרות בצורה טורית, כולל סיבית התחלה וסיביות סיום.
TXRDY - (Transmitter ReaDY) - קו שבעזרתו מודיע ה-USART על "מוכנותו" לקלוט בית נוסף לשידור. קו זה יורד ל-"0" כאשר ה-USART מקבל בית לשידור ועולה אוטומאטית ל-"1" כאשר מוכן לקלוט בית נוסף. העליה ל-"1" קורית בתוך תהליך השידור. ניתן לחבר קו זה כקו פסיקה ל-CPU.
TXE - (Transmitter Empty) - קו שבעזרתו מציין ה-UASRT שהסתיים שידור הבית וה-Transmitter ריק. בדרך כלל, נעזרים בקו זה כאשר התקשורת היא ב-Half Duplex, כלומר, המשדר והקולט משתמשים בערוץ תקשורת אחד. קו זה מסמן שערוץ התקשורת פנוי והקולט יכול לעבור לשידור.
TXC - (Transmitter Clock) - לכניסה זו מספקים גל ריבועי שתדרו קובע את קצב השידור. ניתן לקבוע בתוכנה (כפי שנראה בהמשך), שקצב השידור יהיה בתדר TXC, או 1/16

או 1/64 של TXC.   בדרך כלל, מקובל שתדר TXC יהיה פי 16 מקצב השידור (baud rate).

<u>אותות הקליטה:</u>

<u>RXD</u> - (Receiver Data) - בקו זה מגיע המידע הטורי ל-USART המומר למלה מקבילית בהתאם לאופן העריכה, המתואר בהמשך.

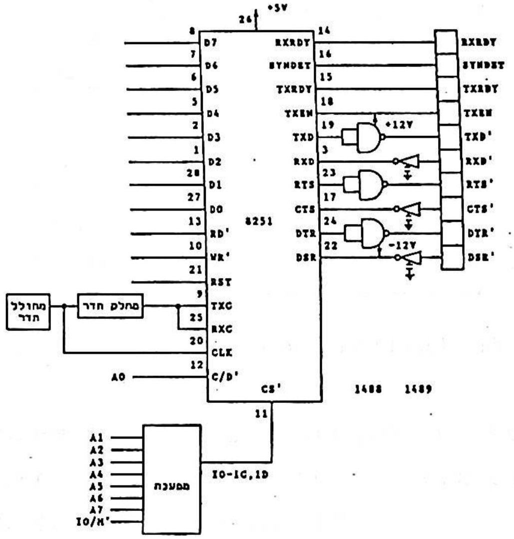
<u>RXRDY</u> - (Receiver ReaDY) - כאשר קו זה עולה ל-"1", הוא מסמן שנקלטה מלה, ויש לקרוא אותה. ניתן לחבר קו זה לקו פסיקה לשם כך.

<u>RXC</u> - (Receiver Clock) - קו זה קובע את קצב קליטת המידע הטורי. גם כאן ניתן בעזרת מלת בקרה לקבוע את קצב הקליטה כ- 1/1, 1/16 או 1/64 של תדר ה-RXC. ניתן לקבוע קצב קליטה שונה מקצב השידור על-ידי מתן אותות שונים ל-TXC ול-RXC, אבל זה לא רגיל.

<u>SYNDET/BRKDET</u> - (SYNc DETect/BReaK DETect) - בעבודה סינכרונית של ה-USART, נקרא קו זה SYNDET. בעבודה אסינכרונית, נקרא הקו BRKDET. בעבודה אסינכרונית עולה קו זה ל-"1", כאשר נקלט נתון שכל הסיביות בו (כולל סיבית הביקורת וסיביות הסיום) הן "0". מצב זה קורה כאשר קו התקשורת מרותק ל-"0".

8.6.2 אופן חיבור ה-8251:

נתאר מעגל חיבור ה-8251 לתקשורת טורית אסינכרונית, כאשר הוא תופס את הכתובות IO-1C,1D.

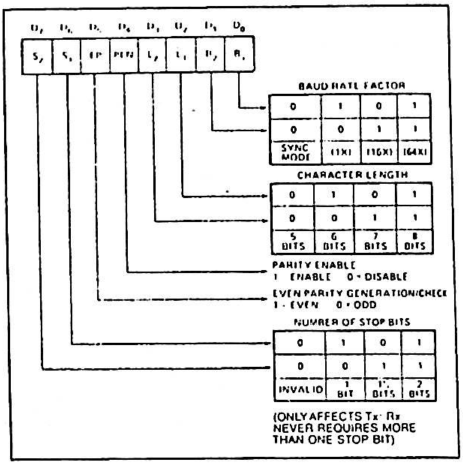


ציור 8-11 מעגל פענוח ל-8251

8.6.3 תכנות ה-8251:

כאמור, תופס ה-8251 שתי כתובות, הנבחרות בעזרת קו A0 שמחובר לקו C/D'. כאשר A0=1, הפניה היא לאוגר COMMAND/ STATUS. בכתיבה, הפניה היא לאוגר COMMAND ובקריאה הפניה היא לאוגר STATUS. כאשר A0=0, הפניה היא לאוגר נתונים לשם מסירת נתון ל-USART או קליטת נתון ממנו.

היתחול ה-USART נעשה על-ידי כתיבת נתונים לאוגר ה-COMMAND. המלה הראשונה המגיעה לאוגר ה-COMMAND, לאחר RESET של ה-8251, קובעת את אופן עבודתו. מלה זו נקראת Mode Instruction.



ציור 8-12 תיאור Mode Instruction

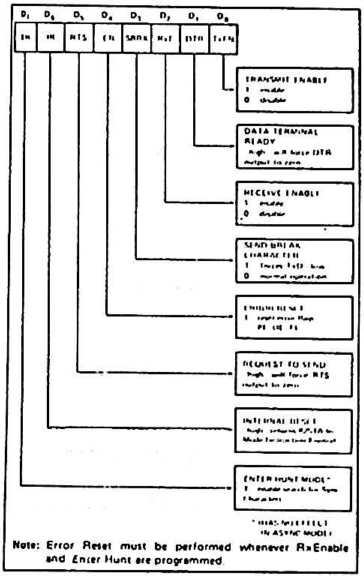
שתי הסיביות הראשונות (B2,B1) קובעות את יחס החלוקה בין RXC ו-TXC לבין קצב השידור והקליטה (X64,X16,X1), או עבודה בצורה סינכרונית (B2,B1=00).
שתי הסיביות (L2,L1) D2 ו-D3 קובעות את אורך התו. כלומר, מספר סיביות המידע. מלה בת 7,6,5 או 8 סיביות.

סיבית (PEN) D4 קובעת אם יש התייחסות לסיבית הביקורת. כלומר, האם נשלחת סיבית ביקורת בתום סיביות המידע או נבדקת סיבית הביקורת בזמן הבדיקה.
PEN=0 - אין התייחסות.
PEN=1 - יש התייחסות.

סיבית (EP) D5 קובעת האם סיבית הביקורת היא סיבית זוגיות או סיבית פרט. EP=0 - פרט, EP=1 - זוגיות.

סיביות (S2,S1) D7,D6 קובעות את מספר סיביות הסיום (1, 1.5 או 2). בדוק מה תבצע מלת Mode Instruction, שערכה EEH.

המלים הבאות, הנכתבות באוגר ה-COMMAND נקראות Command Instruction. הן מפעילות את הרכיב.



ציור 8-13 תיאור Command Instruction

(TXEN) D0 מאפשרת שידור נתונים (Enable-1, Disable-0).

(DTR) D1 מאפשרת אילוץ קו DTR ל-"0". כלומר, מאשר אוטומאטית לרכיב ההיקפי לשדר. אילוץ זה נעשה כאשר כותבים "1" בסיבית זו.

(RXE) D2 מאפשרת קליטת נתונים (Enable-1, Disable-0).

(SBRK) D3 מאפשרת העברת קו TXD למצב BREAK. כלומר, להיות במצב 0" קבוע. "0" בסיבית זו גורם לרכיב לתפקד רגיל.

(ER) D4 מאפסת את הדגלים המציינים שגיאה והמופיעים במלת ה-STATUS, שמתוארת בהמשך. איפוס זה נעשה על-ידי מתן

"1" לסיבית זו.

(RTS) D5 מאפשרת אילוץ קו RTS ל-"0". אילוץ זה נעשה כאשר סיבית זו ב-"1".

(IR) D6 מבצעת RESET בתוכנה לרכיב. לאחר מלת ה-Command Instruction שבה IR=1, יש לתת מלת Mode Instruction מחדש.

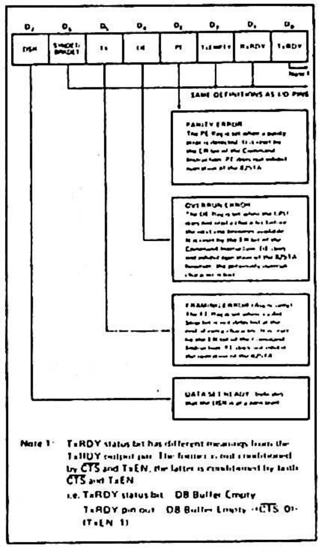
(EH) D7 חסרת משמעות באופן עבודה אסינכרוני.

בדוק מה תבצע מלת Command Instruction שערכה 37H.

לאחר היתחול הרכיב, כל מלה נוספת לאוגר COMMAND, תתפקד כ-Command Instruction. הרכיב מתחיל לתפקד והתקשורת הטורית מתבצעת. אין להתחיל לשדר מיד, אלא לבצע השהייה קצרה.

ה-USART 8251 מיועד לתפקד הן ב-Polling והן בפסיקה. בפסיקה מחברים את TXRDY ו-RXRDY לקוי פסיקה של ה-CPU. רוטינת הפסיקה של השידור משדרת את הנתון הבא. רוטינת הפסיקה של הקליטה אוספת את הנתון.

כדי לבצע קליטה ושידור ב-Polling, קוראים את אוגר ה-STATUS. אוגר זה מציין את מצב ה-USART. אוגר זה נמצא באותה הכתובת כמו אוגר ה-COMMAND (A0=1), רק שהוא מיועד לקריאה ממנו. בקריאה ממנו מתקבל המידע הבא.



ציור 8-14 אוגר Status

TXRDY (D0) מציין את מצב חוצץ השידור. ברגע שחוצץ השידור ריק וה-USART מוכן לקליטת נתון נוסף לשידור, עולה סיבית זו ל-1. זה שונה מקו TXRDY שעולה ל-1 רק כאשר החוצץ ריק, CTS=0 ו-TXE=1.

RXRDY (D1) מציין את מצב חוצץ קליטת הנתונים. כאשר RXRDY=1, יש נתון שהגיע בתקשורת בחוצץ.

TXEMPTY (D2) מציין את מצב הקו TXE. כאשר TXEMPTY=1, ה-USART איננו משדר.

PE (D3) מציין Parity Error. הוא עולה ל-1 כאשר במלה שהתקבלה, סיבית הביקורת לא תואמת את הגדרת הביקורת שנקבעה ב-Mode Instruction.

OE (D4) מציין Overrun Error. הוא עולה ל-1 אם ה-CPU לא קרא נתון שהגיע ונעלם כי בינתיים הגיע נתון נוסף, הנמצא בחוצץ הקליטה.

FE (D5) מציין Framing Error. הוא עולה ל-1 כאשר לא התקבלו בסוף המלה סיביות סיום כמוגדר ב-Mode Instruction.

SYNDET (D6) מציין את מצב הקו SYNDET.

DSR (D7) עולה ל-1 כאשר קו DSR ב-0.

בדרך כלל, מתייחסים לסיביות TXRDY ו-RXRDY בעיקר, כאשר עובדים ב-Polling. נתאר תוכנית המבצעת היתחול ל-8251, קולטת נתון בתקשורת RS232 ממסוף ומשדרת אותו בחזרה למסוף בלולאה אינסופית. פעולה זו נקראת ECHO (הד). אין התייחסות לסיביות הבקרה RTS, CTS, DTR, DSR.

8.7 שאלות ותרגילים

8.7.1 ברשותך 2 פורטי מוצא ופורט מבוא אחד. 8 סיביות פורט מוצא מחוברים לקוי כניסות הנתונים של המדפסת. סיבית D0 של פורט מוצא 2 מחוברת לקו STB של המדפסת. סיבית D0 של פורט המבוא מחוברת לקו BUSY של המדפסת.

א) שרטט את אופן החיבור בין הפורטים למדפסת.

ב) נתון בלוק נתונים בצופן ASCII המסתיים במספר 00 כזקיף. רשום אלגוריתם לשידור ולהדפסת בלוק נתונים זה במדפסת בעזרת המעגל המתואר, כמתואר בסעיף 8.2.

ג) פורט מוצא 1 ופורט המבוא מחוברים לכתובת IO-11. פורט מוצא 2 מחובר לכתובת IO-12. המר את האלגוריתם מסעיף ב' לתוכנית, בהתאם לסוג המיקרופרוססור שברשותך.

8.7.2 רשום תוכנית ראשית ותוכנית פסיקה לשידור בלוק נתונים עבור המעגל המתואר בציור 3-8.

8.7.3 הסבר בקצרה את תפקידי הקוים בתקשורת RS232.

8.7.4 רוצים לשדר ב-USART בקצב של baud 4800. משתמשים בחלוקת ה-CLK ל-16.

א) מהו תדר ה-CLK הנדרש.

ב) ה-USART כיזון מיציאת מונה 0 של ה-8254. מונה זה מקבל אות בתדר של 6.144MHz. רשום פקודות היתחול ל-8254 כדי לקבל בכניסת ה-USART את התדירות הדרושה.

8.7.5 ה-USART מחובר כ-DTE למחשב בתקשורת RS232, בעזרת 3 גידים בלבד RXD, TXD ו-GND.

רשום תוכנית המבצעת:

א) היתחול ל-8254 כך שנקבל CLK בתדר 76,800Hz (ראה שאלה 8.7.4).

ב) היתחול ל-8251 לתקשורת טורית אסינכרונית לפי 1 סיבית התחלה, 8 סיביות נתון, ללא סיבית ביקורת, 2 סיביות סיום.

אין התייחסות לקוים CTS, DSR.

ג) המתנה לתו מ-RXD ב-Polling.

ד) שידור תו דרך TXD ב-Polling.

ה) חזרה לסעיף ג'.

8.7.6 רשום תוכנית המבצעת:

א) היתחול בדומה לתרגיל 8.7.5.

ב) ממתינה לתו מלוח המקשים (ראה רוטינות עזר בספר המשתמש של הערכה שברשותך).

ג) משדרת את התו ב-Polling.

ד) חוזרת לסעיף ב'.

8.7.7 רשום תוכנית המבצעת:

א) היתחול בדומה לתרגיל 8.7.5.

ב) ממתינה לתו מהתקשורת הטורית.

ג) מוציאה את התו לנוריות IO-00.

ד) חוזרת לסעיף ב'.

8.7.8 רשום פקודות היתחול בדומה לתרגיל 8.7.5, רק עם התייחסות לקוים CTS ו-DSR.

8.8 תקשורת מקבילית למדפסת

כאשר קובעים נוהל תקשורת בין מחשבים קובעים גם את צורת החיבור. משתמשים במחברים סטנדרטיים וקובעים תפקיד לכל רגל. נתייחס לחלק מהקוים של התקשורת וניצור תקשורת אל מדפסת.

אם נתבונן במחבר הנמצא בקצהו של כבל מדפסת (בצד של המדפסת) נראה שלכל רגל יש מספר (או לפחות מסומנת רגל 1). את המספר הבינארי המציין את התו המשודר שולחים דרך הרגליים 2 עד 9, כאשר D0 היא רגל 2 ו-D7 היא רגל 9. רגל 1 משמשת כקו תזמון (Strobe). דרך קו זה מודיעים למדפסת שמידע חדש ממתין בקוים 2 עד 9. הודעה זו נעשית על-ידי הורדת הקו ל-"0" והעלאתו ל-"1". את הקוים 1 עד 9 מחברים לקוים של פורטי מוצא.

משני צידי כבל המדפסת יש שני מחברים. מחבר גדול, המתחבר למדפסת ומחבר קטן יותר המתחבר למחשב.

ב-KIT הרכיבים תמצא כבל רב גידי שמחבר בן 25 פינים בקצהו. מחבר זה יש לחבר למחבר הקטן יותר (המתאים לו) של כבל המדפסת. את המחבר הגדול יש לחבר למדפסת.

כל גיד של הכבל הרב גידי הוא בצבע שונה והם מחוברים לרגליים של המחבר בצורה הבאה:

רגל 1 - חום.
רגל 2 - אדום.
רגל 3 - כתום.
רגל 4 - צהוב.
רגל 5 - ירוק.
רגל 6 - כחול.
רגל 7 - סגול.
רגל 8 - אפור.
רגל 9 - לבן.
רגל 11 - ורוד.
רגל 18 - שחור.

למדפסת לוקח זמן לבצע את פעולת ההדפסה. זמן זה גם איננו אחיד. אם למשל, היא מקבלת הודעה: "כנע לעמוד חדש", הפעולה תקח זמן ארוך יחסית להדפסת תו בודד. כדי להתגבר על בעיה זו, מחברים את רגל 11 לפורט מבוא. קו זה נקרא קו Busy (עסוק). דרך קו זה, מודיעה המדפסת אם היא מסוגלת לקלוט תו או שהיא עסוקה בטפול בתוים קודמים.

<u>אלגוריתם התקשורת יהיה אם כך כדלקמן:</u>

א. בדיקה, האם קו 11 (Busy) נמצא ב-"1". אם לא, המתנה עד שיעלה ל-"1".
ב. מסירה לקוים 2 עד 9 את צופן התו המבוקש להדפסה.
ג. הורדת קו 1 (Strobe) ל-"0" והעלאתו ל-"1".

8.8.1 חבר כבל תקשורת למדפסת אל יציאות APPLIC-5 בצורה הבאה:

א) את קו 1 (חום) חבר ליציאה Q7.
ב) את קוים 2 עד 8 חבר ליציאות Q0 עד Q6.
ג) את קו 9 (לבן) חבר ל-GND. אנו נשלח מספרים בינאריים בני שבע סיביות בלבד בהתאם לצופן ASCII.
ד) את קו 11 (ורוד-Busy) חבר לכניסה J0.
ה) את קו 18 (שחור-GND) חבר ל-GND.

רשום תוכנית המשדרת בלוק תוים למדפסת. את בלוק התוים רשום בשתי שורות של תוי אותיות, כאשר בסוף כל שורה יש את תוי הבקרה 0D (CR - Carriage Return) ו-0A (LF - Line Feed). אם הנך מתקשה, היעזר בתרגיל הפתור שבסוף הפרק.

8.9 תקשורת טורית אסינכרונית חלקית

בתרגול זה נבצע תקשורת בין שתי ערכות. בקיט הרכיבים המצורף למעגל DSM-2200 יש בכל תקשורת מסוכך בן 3 גידים (אחד הגידים הוא הסיכוך).

8.9.1 וודא חיבור כרטיס DSM-2200 לערכה S-2100 בשתי הערכות.

8.9.2 חבר חוט הסיכוך של כבל התקשורת ל-GND בשני המעגלים.

8.9.3 חבר את החוט האדום בערכה א' ל-TXD ובערכה ב' ל-RXD.

8.9.4 חבר את החוט השחור בערכה א' ל-RXD ובערכה ב' ל-TXD.

8.9.5 רשום בערכה א' תוכנית, כמתואר בשאלה 8.7.6.

8.9.6 רשום בערכה ב' תוכנית, כמתואר בשאלה 8.7.7.

8.9.7 הרץ את שתי התוכניות.

8.9.8 לחץ על מקשים בערכה א'. מספריהם צריכים להופיע בנוריות שבערכה ב'.

8.10 <u>תקשורת טורית אסינכרונית מלאה</u>

8.10.1 שנה בתוכניות שבשתי הערכות את פקודות ההיתחול כך שתהיה התייחסות ל-RTS ול-DTR.

8.10.2 הוסף חוטי קישור בין ארבע הנקודות בהתאם.

<u>ערכה א'</u>		<u>ערכה ב'</u>
RTS	⟶	DTR
CTS	⟵	DSR

מספיק חיבור של שני חוטים, מכיוון שרק ערכה א' משדרת לערכה ב'. אם גם ערכה ב' היתה צריכה לשדר, היינו צריכים להוסיף שני חוטים נוספים.

8.10.3 הרץ את שתי התוכניות.

8.10.4 לחץ על מקשים בערכה א'. מספריהם צריכים להופיע בנוריות שבערכה ב'.

8.10.5 נתק את החוט המגשר בין DSR ל-CTS. בדוק אם המערכות מגיבות כראוי.

8.10.6 שנה את תוכנית ערכה ב', כך שלאחר שקלטה תו בתקשורת הטורית היא משדרת אותו בחזרה.

8.10.7 שנה את תוכנית ערכה א', כך שבמקביל לבדיקת לוח המקשים היא בודקת אם ממתין תו ב-USART. אם ממתין תו ב-USART, היא אוספת אותו ומוציאה אותו לנוריות הערכה. אם נלחץ מקש היא משדרת אותו לערכה ב'.

8.10.8 הרץ את התוכניות.

8.10.9 לחיצה על מקש בערכה א' צריכה להדליק בנוריות ערכה א' את מספר המקש, שמשודר לערכה ב' וחוזר ממנה.

<u>תרגיל פתור:</u>

בלוק הנתונים (שורות TABLE) משותף לשלושת הפתרונות.

<u>8085</u>

```
PRINT:  LXI  H,TABLE
PRINT2: MOV  B,M
        MOV  A,B
        CPI  00
        JZ   PRINT4
PRINT3: IN   59
        ANI  01
        JZ   PRINT3
        MOV  A,B
        ADI  80
        OUT  59
        MOV  A,B
        OUT  59
        INX  H
        JMP  PRINT2
;
PRINT4: RST  1
;
;
TABLE:
```

<u>8051</u>

```
PRINT:  MOV  DPTR,#TABLE
PRINT2: MOVX A,@DPTR
        MOV  R0,A
        CJNE R0,#0,PRINT3
        LJMP 0000
;
PRINT3: PUSH DPH
        PUSH DPL
        MOV  DPTR,#FF59
PRINT4: MOVX A,@DPTR
        JNB  ACC.0,PRINT4
        SETB ACC.7
        MOVX @DPTR,A
        MOV  A,R0
        MOVX @DPTR,A
        POP  DPL
        POP  DPH
        INC  DPTR
        SJMP PRINT2
TABLE:  DB   53,45,53,0D,0A
        DB   41,50,50,4C,49,43,0D,0A,00
```

<u>8086</u>

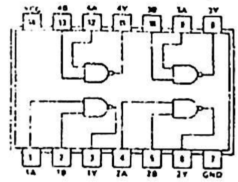
```
PRINT:  MOV  BX,TABLE
PRINT2: MOV  AH,[BX]
        CMP  AH,00
        JZ   PRINT4
        MOV  DX,359
PRINT3: IN   AL,DX
        AND  AL,1
        JZ   PRINT3
        MOV  AL,AH
        ADD  AL,80
        OUT  DX,AL
        MOV  AL,AH
        OUT  DX,AL
        INC  BX
        JMP  PRINT2
PRINT4: INT  3
;
TABLE:
```

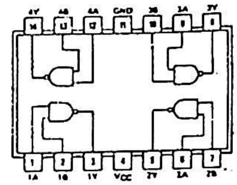
QUADRUPLE 2-INPUT
POSITIVE-NAND GATES

00

positive logic:
Y = $\overline{AB}$

See page 6-2





SN5400 (J)
SN54H00 (J)
SN54L00 (J)
SN54LS00 (J, W)
SN54S00 (J, W)

SN7400 (J, N)
SN74H00 (J, N)
SN74L00 (J, N)
SN74LS00 (J, N)
SN74S00 (J, N)

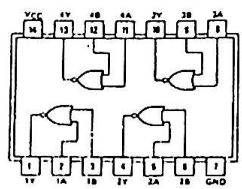
SN5400 (W)
SN54H00 (W)
SN54L00 (T)

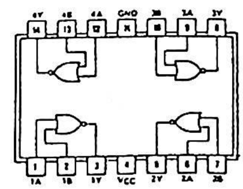
QUADRUPLE 2-INPUT
POSITIVE-NOR GATES

02

positive logic:
Y = $\overline{A+B}$

See page 6-8





SN5402 (J)
SN54L02 (J)
SN54LS02 (J, W)
SN54S02 (J, W)

SN7402 (J, N)
SN74L02 (J, N)
SN74LS02 (J, N)
SN74S02 (J, N)

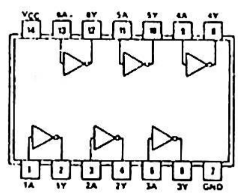
SN5402 (W)
SN54L02 (T)

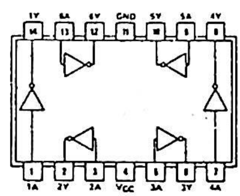
HEX INVERTERS
WITH OPEN-COLLECTOR OUTPUTS

05

positive logic:
Y = $\overline{A}$

See page 6-4





SN5405 (J)
SN54H05 (J)
SN54LS05 (J, W)
SN54S05 (J, W)

SN7405 (J, N)
SN74H05 (J, N)
SN74LS05 (J, N)
SN74S05 (J, N)

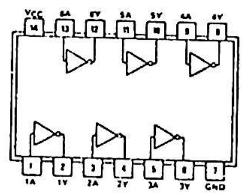
SN5405 (W)
SN54H05 (W)

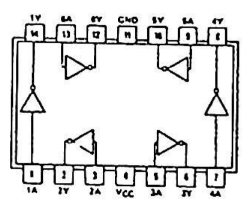
HEX INVERTERS

04

positive logic:
Y = $\overline{A}$

See page 6-2





SN5404 (J)
SN54H04 (J)
SN54L04 (J)
SN54LS04 (J, W)
SN54S04 (J, W)

SN7404 (J, N)
SN74H04 (J, N)
SN74L04 (J, N)
SN74LS04 (J, N)
SN74S04 (J, N)

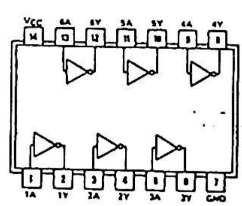
SN5404 (W)
SN54H04 (W)
SN54L04 (T)

HEX INVERTER BUFFERS/DRIVERS
WITH OPEN-COLLECTOR
HIGH-VOLTAGE OUTPUTS

06

positive logic:
Y = $\overline{A}$

See page 6-24



SN5406 (J, W)

SN7406 (J, N)

QUADRUPLE BUS BUFFER GATES WITH THREE STATE OUTPUTS

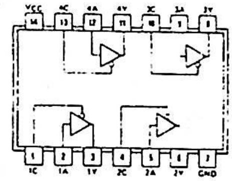
125

positive logic:

Y = A

Output is off (disabled) when C is high.

See page 6-33



SN54125 (J, W) SN74125 (J, N)
SN54LS125A (J, W) SN74LS125A (J, N

QUADRUPLE BUS BUFFER GATES WITH THREE STATE OUTPUTS

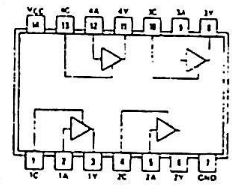
126

positive logic:

Y = A

Output is off (disabled) when C is low

See page 6-33

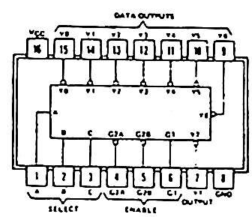


SN54126 (J, W) SN74126 (J, N)
SN54LS126A (J, W) SN74LS126A (J, N

3-TO-8 LINE DECODERS/MULTIPLEXERS

138

See page 7-134

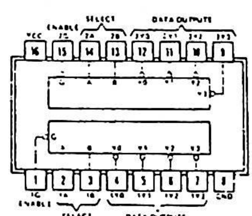


SN54LS138 (J, W) SN74LS138 (J, N)
SN54S138 (J, W) SN74S138 (J, N)

DUAL 2-TO-4 LINE DECODERS/MULTIPLEXERS

139

See page 7-134



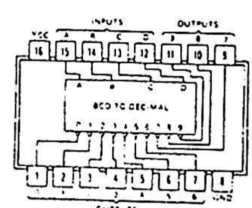
SN54LS139 (J, W) SN74LS139 (J, N)
SN54S 139 (J, W) SN74S139 (J, N)

BCD-TO-DECIMAL DECODERS/DRIVERS FOR LAMPS, RELAYS, MOS

145

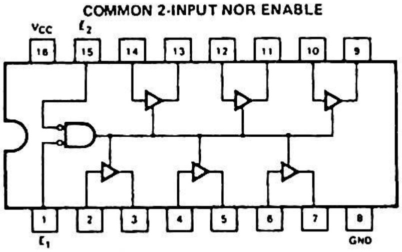
BCD-TO DECIMAL

See page 7 148



SN54145 (J, W) SN74145 (J, N)
SN54LS145 (J, W) SN74LS145 (J, W)

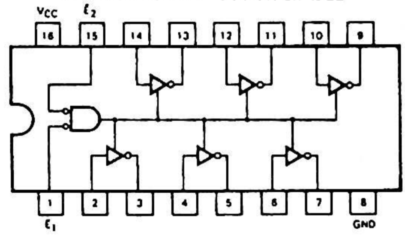
SN54LS/74LS365A
HEX 3-STATE BUFFER WITH COMMON 2-INPUT NOR ENABLE



TRUTH TABLE

INPUTS			OUTPUT
$\bar{E}_1$	$\bar{E}_2$	D	
L	L	L	L
L	L	H	H
H	X	X	(Z)
X	H	X	(Z)

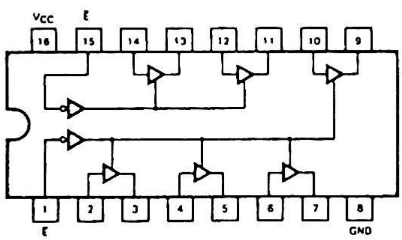
SN54LS/74LS366A
HEX 3-STATE INVERTER BUFFER WITH COMMON 2-INPUT NOR ENABLE



TRUTH TABLE

INPUTS			OUTPUT
$\bar{E}_1$	$\bar{E}_2$	D	
L	L	L	H
L	L	H	L
H	X	X	(Z)
X	H	X	(Z)

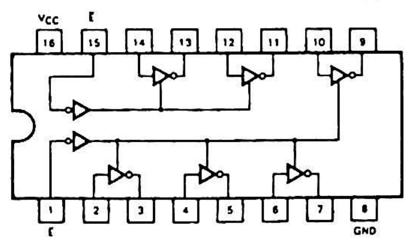
SN54LS/74LS367A
HEX 3-STATE BUFFER SEPARATE 2-BIT AND 4-BIT SECTIONS



TRUTH TABLE

INPUTS		OUTPUT
$\bar{E}$	D	
L	L	L
L	H	H
H	X	(Z)

SN54LS/74LS368A
HEX 3-STATE INVERTER BUFFER SEPARATE 2-BIT AND 4-BIT SECTIONS

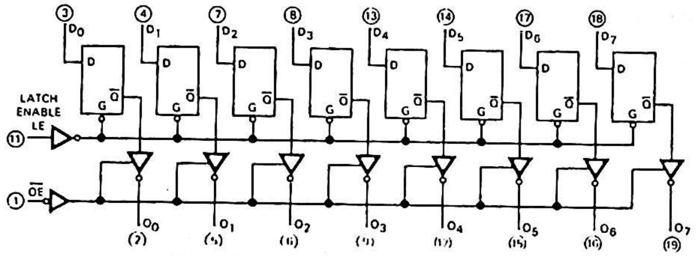


TRUTH TABLE

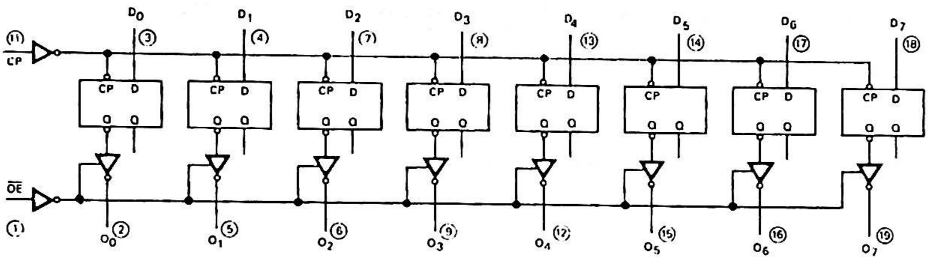
INPUTS		OUTPUT
$\bar{E}$	D	
L	L	H
L	H	L
H	X	(Z)

LOGIC DIAGRAMS

SN54LS/74LS373



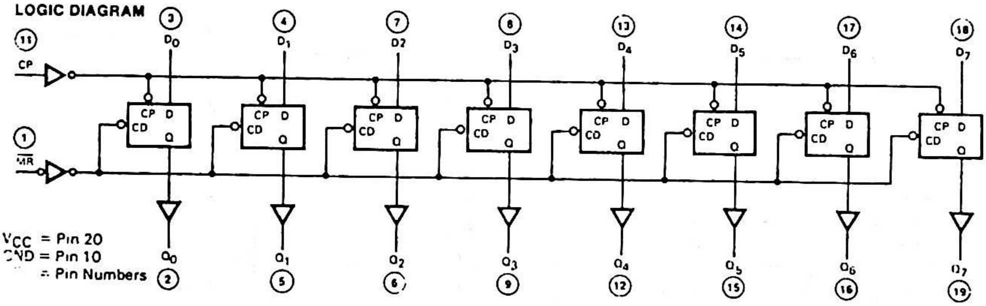
SN54LS/74LS374



V_{CC} Pin 20
GND = Pin 10
○ = Pin Numbers

SN54LS/74LS273

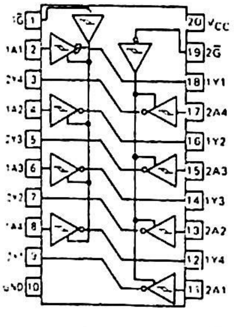
LOGIC DIAGRAM



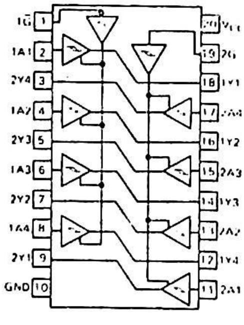
V_{CC} = Pin 20
GND = Pin 10
○ = Pin Numbers

LOGIC AND CONNECTION DIAGRAMS DIP (TOP VIEW)

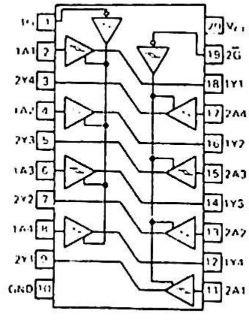
SN54LS/74LS240



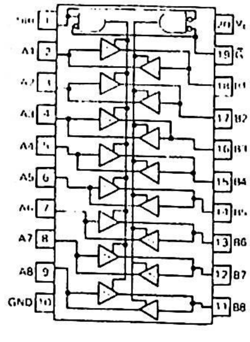
SN54LS/74LS241

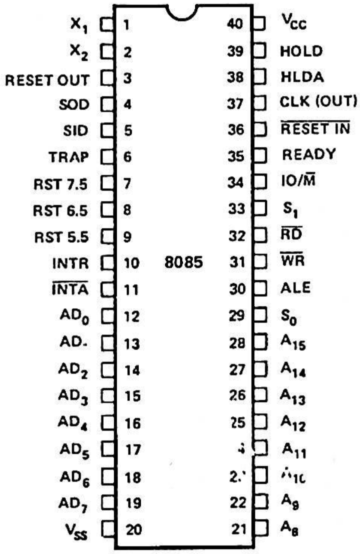


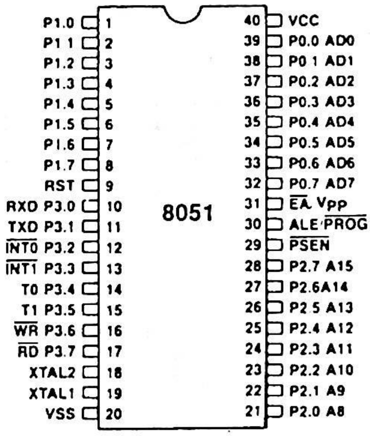
SN54LS/74LS244

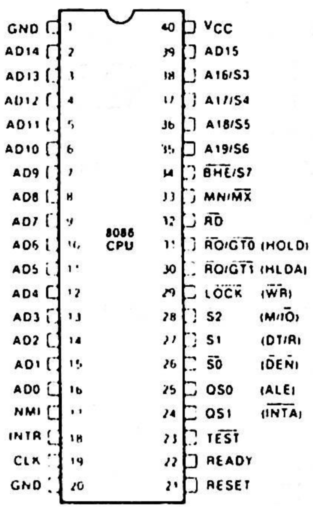


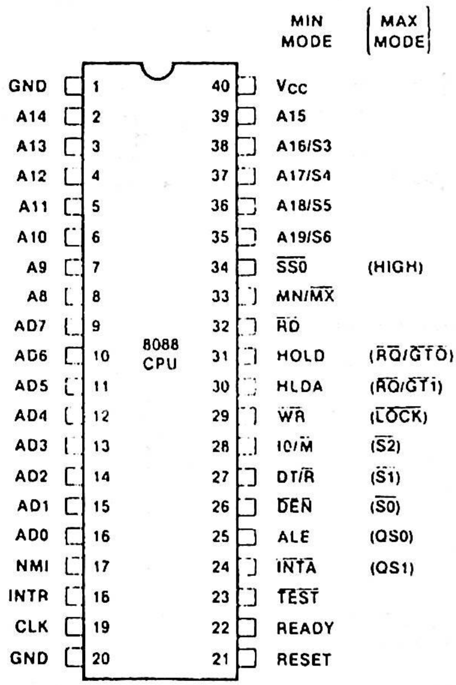
SN54LS/74LS245



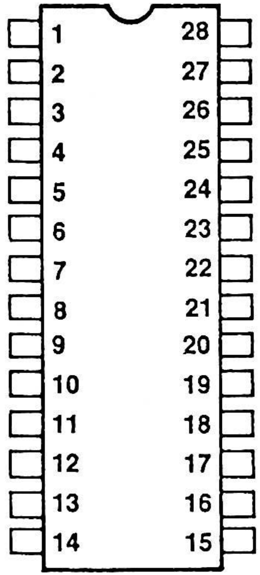

8085
X1 1
X2 2
RESET OUT 3
SOD 4
SID 5
TRAP 6
RST 7.5 7
RST 6.5 8
RST 5.5 9
INTR 10
INTA 11
AD0 12
AD1 13
AD2 14
AD3 15
AD4 16
AD5 17
AD6 18
AD7 19
VSS 20
40 VCC
39 HOLD
38 HLDA
37 CLK (OUT)
36 RESET IN
35 READY
34 IO/M
33 S1
32 RD
31 WR
30 ALE
29 S0
28 A15
27 A14
26 A13
25 A12
A11
A10
22 A9
21 A8


8051
P1.0 1
P1.1 2
P1.2 3
P1.3 4
P1.4 5
P1.5 6
P1.6 7
P1.7 8
RST 9
RXD P3.0 10
TXD P3.1 11
INT0 P3.2 12
INT1 P3.3 13
T0 P3.4 14
T1 P3.5 15
WR P3.6 16
RD P3.7 17
XTAL2 18
XTAL1 19
VSS 20
40 VCC
39 P0.0 AD0
38 P0.1 AD1
37 P0.2 AD2
36 P0.3 AD3
35 P0.4 AD4
34 P0.5 AD5
33 P0.6 AD6
32 P0.7 AD7
31 EA VPP
30 ALE PROG
29 PSEN
28 P2.7 A15
27 P2.6 A14
26 P2.5 A13
25 P2.4 A12
24 P2.3 A11
23 P2.2 A10
22 P2.1 A9
21 P2.0 A8


8086 CPU
GND 1
AD14 2
AD13 3
AD12 4
AD11 5
AD10 6
AD9 7
AD8 8
AD7 9
AD6 10
AD5 11
AD4 12
AD3 13
AD2 14
AD1 15
AD0 16
NMI 17
INTR 18
CLK 19
GND 20
40 VCC
39 AD15
38 A16/S3
37 A17/S4
36 A18/S5
35 A19/S6
34 BHE/S7
33 MN/MX
32 RD
31 RQ/GT0 (HOLD)
30 RQ/GT1 (HLDA)
29 LOCK (WR)
28 S2 (M/IO)
27 S1 (DT/R)
26 S0 (DEN)
25 QS0 (ALE)
24 QS1 (INTA)
23 TEST
22 READY
21 RESET


MIN MODE
MAX MODE
8088 CPU
GND 1
A14 2
A13 3
A12 4
A11 5
A10 6
A9 7
A8 8
AD7 9
AD6 10
AD5 11
AD4 12
AD3 13
AD2 14
AD1 15
AD0 16
NMI 17
INTR 18
CLK 19
GND 20
40 VCC
39 A15
38 A16/S3
37 A17/S4
36 A18/S5
35 A19/S6
34 SS0 (HIGH)
33 MN/MX
32 RD
31 HOLD (RQ/GT0)
30 HLDA (RQ/GT1)
29 WR (LOCK)
28 IO/M (S2)
27 DT/R (S1)
26 DEN (S0)
25 ALE (QS0)
24 INTA (QS1)
23 TEST
22 READY
21 RESET

Pin	6264	27256	27128	2764	2732A	2716
1	N.C.	V_{PP}	V_{PP}	V_{PP}		
2	A_{12}	A_{12}	A_{12}	A_{12}		
3	A_7	A_7	A_7	A_7	A_7	A_7
4	A_6	A_6	A_6	A_6	A_6	A_6
5	A_5	A_5	A_5	A_5	A_5	A_5
6	A_4	A_4	A_4	A_4	A_4	A_4
7	A_3	A_3	A_3	A_3	A_3	A_3
8	A_2	A_2	A_2	A_2	A_2	A_2
9	A_1	A_1	A_1	A_1	A_1	A_1
10	A_0	A_0	A_0	A_0	A_0	A_0
11	D_0	O_0	O_0	O_0	O_0	O_0
12	D_1	O_1	O_1	O_1	O_1	O_1
13	D_2	O_2	O_2	O_2	O_2	O_2
14	Gnd	Gnd	Gnd	Gnd	Gnd	Gnd



Pin	2716	2732A	2764	27128	27256	6264
28			V_{CC}	V_{CC}	V_{CC}	V_{CC}
27			$\overline{PGM}$	$\overline{PGM}$	A_{14}	$\overline{WR}$
26	V_{CC}	V_{CC}	N.C.	A_{13}	A_{13}	CE
25	A_8	A_8	A_8	A_8	A_8	A_8
24	A_9	A_9	A_9	A_9	A_9	A_9
23	V_{PP}	A_{11}	A_{11}	A_{11}	A_{11}	A_{11}
22	$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$
21	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
20	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
19	O_7	O_7	O_7	O_7	O_7	D_7
18	O_6	O_6	O_6	O_6	O_6	D_6
17	O_5	O_5	O_5	O_5	O_5	D_5
16	O_4	O_4	O_4	O_4	O_4	D_4
15	O_3	O_3	O_3	O_3	O_3	D_3